|  |  |
| --- | --- |
| ГОСУДарственный стандарт Республики Беларусь | СТБ/ПР\_1/  (ISO/IEC/IEEE 12207) |
|  |  |

**СИСТЕМНАЯ И ПРОГРАММНАЯ ИНЖЕНЕРИЯ**

**ПРОЦЕССЫ ЖИЗНЕННОГО ЦИКЛА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ**

**CIСТЭМНАЯ I ПРАГРАМНАЯ IНЖЫНЕРЫЯ**

**ПРАЦЭСЫ ЖЫЦЦЕВАГА ЦЫКЛА ПРАГРАМНАГА ЗАБЯСПЯЧЭННЯ**

**(ISO/IEC/IEEE 12207, IDT)**

*Настоящий проект стандарта не подлежит применению до его принятия*



**Госстандарт**

**Минск**

УДК МКС 33.040 КП

**Ключевые слова:** приобретение, соглашение, архитектура, дизайн, оценка, аудит, управление конфигурацией, управление решениями, разработка, утилизация, вспомогательная система, внедрение, управление информацией, инфраструктура, интеграция, жизненный цикл, модель жизненного цикла, этапы жизненного цикла, обслуживание, измерение, эксплуатация, планирование, процесс, улучшение процесса, эталонная модель процесса, адаптация процесса, представление процесса, продукт, портфель, управление качеством, требования, выход из эксплуатации, управление рисками, услуга, программное обеспечение, программная система, интересующая система, этапы, требования заинтересованных сторон, поставка, система, структура системы, интересующая система, адаптация, переход, валидация, верификация.

**Предисловие**

Цели, основные принципы, положения по государственному регулированию и управлению в области технического нормирования и стандартизации установлены Законом Республики Беларусь   
«О техническом нормировании и стандартизации».

1 РАЗРАБОТАН открытым акционерным обществом «Гипросвязь» (ОАО «Гипросвязь»)

2 УТВЕРЖДЕН И ВВЕДЕН В ДЕЙСТВИЕ постановлением Госстандарта Республики Беларусь   
от №

3 Настоящий стандарт идентичен по отношению к международному стандарту ISO/IEC/IEEE 12207 «Systems and soft engineering – Software life cycle processes».

Международный стандарт разработан Institute of Electrical and Electronics Engineers, Inc.

Степень соответствия – идентичная (IDT).

4 ВВЕДЕН ВПЕРВЫЕ

Настоящий стандарт не может быть воспроизведен, тиражирован и распространен в качестве официального издания без разрешения Госстандарта Республики Беларусь

Издан на русском языке

Содержание

Предисловие

Введение

1 Сфера действий 1

1.1 Общие сведения 1

1.2 Цель 1

1.3 Область применения 1

1.4 Ограничения 2

2 Нормативные ссылки 2

3 Термины, определения и сокращения 2

3.1 Термины, определения 2

3.2 Сокращения 11

4 Соответствие 11

4.1 Предполагаемое использование 11

4.2 Полное соответствие 12

4.2.1 Полное соответствие результатам 12

4.2.2 Полное соответствие задачам 12

4.3 Индивидуальное соответствие 12

5 Ключевые концепции и применение 13

5.1 Введение 13

5.2 Концепции программных систем 13

5.2.1 Программные системы 13

5.2.2 Структура программной системы 13

5.2.3 Обеспечивающие системы 15

5.2.4 Процессы жизненного цикла для программной системы 16

5.3 Организация и концепции проекта 16

5.3.1 Организации 16

5.3.2 Внедрение на уровне организации и проектов 17

5.4 Концепции жизненного цикла 17

5.4.1 Этапы жизненного цикла программного обеспечения 17

5.4.2 Модель жизненного цикла программной системы 17

5.5 Концепции процесса 19

5.5.1 Критерии процессов 19

5.5.2 Описание процессов 19

5.5.3 Общие характеристики процессов 19

5.5.4 Адаптация 19

5.6 Группы процессов 19

5.6.1 Введение 19

5.6.2 Процессы согласования 21

5.6.3 Организационные процессы поддержки проектов 22

5.6.4 Процессы технического управления 22

5.6.5 Технические процессы 22

5.7 Применение процесса 22

5.8 Эталонная модель процесса 23

6 Процессы жизненного цикла программного обеспечения 24

6.1 Процессы согласования 24

6.1.1 Процесс приобретения 24

6.1.2 Процесс поставки 27

6.2 Организационные процессы поддержки проекта 28

6.2.1 Процесс управления моделью жизненного цикла 29

6.2.2 Процесс управления инфраструктурой 30

6.2.3 Процесс управления проектным портфелем 31

6.2.4 Процесс управления человеческими ресурсами 33

6.2.5 Процесс менеджмента качества 34

6.2.6 Процесс управления знаниями 36

6.3 Процессы технического менеджмента 37

6.3.1 Процесс планирования проекта 38

6.3.2 Процесс оценки и контроля проекта 40

6.3.3 Процесс управления решениями 43

6.3.4 Процесс управления рисками 44

6.3.5 Процесс управления конфигурацией 46

6.3.6 Процесс управления информацией 50

6.3.7 Процесс оценки 52

6.3.8 Процесс обеспечения качества 53

6.4 Технические процессы 55

6.4.1 Процесс анализа бизнеса или цели работы 56

6.4.2 Процесс определения потребностей заинтересованных сторон 59

6.4.3 Процесс определения требований к системе/программному обеспечению 63

6.4.4 Процесс определения архитектуры 66

6.4.5 Процесс определения проектного решения 71

6.4.6 Процесс системного анализа 74

6.4.7 Процесс внедрения 75

6.4.8 Процесс интеграции 79

6.4.9 Процесс верификации 82

6.4.10 Переходный процесс 85

6.4.11 Процесс валидации 89

6.4.12 Рабочий процесс 92

6.4.13 Процесс обслуживания 95

6.4.14 Процесс утилизации 99

Приложение A (обязательное) Процесс адаптации 102

A.1 Введение 102

A.2 Процесс адаптации 102

A.2.1 Цель 102

A.2.2 Результаты 102

A.2.3 Действия и задачи 102

Приложение B (рекомендательное) Примеры информационных элементов процесса 104

Приложение C (рекомендательное) Эталонная модель процесса для целей оценки 107

C.1 Введение 107

C.2 Соответствие ISO/IEC 33004 107

C.2.1 Общая часть 107

C.2.2 Требования к эталонным моделям процессов 107

C.2.3 Описания процессов 108

C.3 Эталонная модель процесса 108

Приложение D (рекомендательное) Интеграция процессов и процессные конструкции 109

D.1 Введение 109

D.2 Конструкции процессов и их использование 109

Приложение E (рекомендательное) Представления процесса 111

E.1 Введение 111

E.2 Концепция представления процесса 111

E.3 Точка обзора процесса 111

E.4 Вид процесса для специализированного инжиниринга 112

E.5 Представление процесса для управления интерфейсом 114

E.6 Представление процесса обеспечения безопасности программного обеспечения

(Информационная безопасность) 116

Приложение F (рекомендательное) Моделирование архитектуры программных систем 120

F.1 Введение 120

F.2 Представления, модели и виды моделей, используемые в архитектуре программных систем 120

F.2.1 Функциональная модель 120

F.2.2 Статическая модель 121

F.2.3 Модель данных 121

F.2.4 Поведенческая модель 121

F.2.5 Временная модель 121

F.2.6 Структурная модель 121

F.2.7 Сетевая модель 121

F.3 Другие соображения по модели 121

Приложение G (рекомендательное) Применение процессов жизненного цикла программного обеспечения к суперсистеме 123

G.1 Введение 123

G.2 Характеристики и типы суперсистем 123

G.3 Процессы системной инженерии, применяемые к суперсистемам 124

G.3.1 Общая часть 124

G.3.2 Процессы согласования 124

G.3.3 Организационные процессы, способствующие реализации проектов 124

G.3.4 Процессы технического менеджмента 125

G.3.5 Технические процессы 125

Приложение H (рекомендательное) Применение Agile 127

Приложение I (рекомендательное) Отображение процессов согласно ISO/IEC/IEEE 12207: 2008 129

Библиография 143

**Предисловие**

ISO (Международная организация по стандартизации) и IEC (Международная электротехническая комиссия) образуют специализированную систему всемирной стандартизации. Национальные органы, являющиеся членами ISO или IEC, участвуют в разработке международных стандартов через технические комитеты, созданные соответствующей организацией для работы в определенных областях технической деятельности. Технические комитеты ISO и IEC сотрудничают в областях, представляющих взаимный интерес. Другие международные организации, правительственные и неправительственные, связанные с ISO и IEC, также принимают участие в работе. В области информационных технологий ISO и IEC создали совместный технический комитет ISO/IEC JTC 1.

Документы по стандартам IEEE разрабатываются обществами IEEE и комитетами по координации стандартов Совета по стандартам IEEE Standards Association (IEEE-SA). IEEE разрабатывает свои стандарты посредством согласованного процесса разработки, одобренного Американским национальным институтом стандартов, который объединяет добровольцев, представляющих различные точки зрения и интересы, для достижения конечного продукта. Волонтеры не обязательно являются членами Института и работают бесплатно. В то время как IEEE управляет процессом и устанавливает правила для обеспечения объективности в процессе разработки консенсуса, IEEE не проводит независимую оценку, тестирование или проверку точности любой информации, содержащейся в его стандартах.

Процедуры, использованные для разработки данного документа, и процедуры, предназначенные для его дальнейшего ведения, описаны в Директивах ISO/IEC, часть 1. В частности, следует отметить различные критерии утверждения, необходимые для различных типов документов. Данный документ был составлен в соответствии с редакционными правилами Директив ISO/IEC, часть 2 (см. www.iso.org/directives).

Обращаем внимание на возможность того, что некоторые элементы данного документа могут быть предметом патентных прав. ISO, IEC и IEEE не несут ответственности за выявление любых или всех таких патентных прав. Подробная информация о любых патентных правах, выявленных в ходе разработки документа, будет приведена во Введении и/или в списке полученных патентных деклараций ISO (см. www.iso.org/patents).

Любое торговое название, используемое в данном документе, является информацией, приведенной для удобства пользователей, и не означает одобрением.

Для объяснения значения конкретных терминов и выражений ИСО, связанных с оценкой соответствия, а также информации о соблюдении ISO принципов Всемирной торговой организации (ВТО) в отношении технических барьеров в торговле (ТБТ) см. URL-адрес www.iso.org/iso/foreword.html.

Этот документ был подготовлен Объединенным техническим комитетом ISO/IEC JTC 1, Информационные технологии, Подкомитетом SC 7, Системная и программная инженерия, в сотрудничестве с Комитетом по стандартам систем и программной инженерии “IEEE Computer Society”, в соответствии с соглашением о сотрудничестве с Партнерской организацией по разработке стандартов между ISO. и IEEE.

Это первое издание ISO/IEC/IEEE 12207 отменяет и заменяет ISO/IEC 12207: 2008 (второе издание), которое было технически пересмотрено.

Изменения в этой версии ISO / IEC / IEEE 12207 были разработаны вместе с соответствующей версией ISO/IEC/IEEE 15288: 2015 «Системная и программная инженерия - Процессы жизненного цикла системы». Целью этих изменений является согласование структур и содержания двух документов, поддерживая при этом требования сообществ инженеров и специалистов по оценке.

Данный документ был разработан с целью:

– обеспечить общую терминологию между редакцией ISO/IEC/IEEE 15288 и ISO/IEC/IEEE 12207;

– где это применимо, предоставить общие имена процессов и структуру процессов для версий ISO / IEC / IEEE 15288 и ISO/IEC/IEEE 12207; и

– позволяют сообществу пользователей развиваться в направлении полностью гармонизированных стандартов, обеспечивая при этом обратную совместимость.

Данная версия предназначена для достижения полностью согласованного представления о процессах жизненного цикла системы и программного обеспечения.

**Введение**

Сложность программных систем выросла до беспрецедентного уровня. Это привело к появлению новых возможностей, но также и к увеличению проблем для организаций, которые создают и используют системы. Эти проблемы существуют на протяжении всего жизненного цикла системы и на всех уровнях архитектурной детализации. В этом документе представлена общая структура процесса для описания жизненного цикла систем, созданных людьми, с использованием подхода программной инженерии. Программная инженерия - это междисциплинарный подход и средство, позволяющее реализовать успешные программные системы. Он фокусируется на определении потребностей заинтересованных сторон и требуемой функциональности на ранних этапах цикла разработки, документировании требований и выполнении синтеза проекта и валидации системы при рассмотрении всей проблемы. Он объединяет все дисциплины и группы специальностей в командные усилия, формируя структурированный процесс разработки, который переходит от концепции к производству, эксплуатации и техническому обслуживанию. Он учитывает как деловые, так и технические потребности всех заинтересованных сторон с целью предоставления качественного продукта, отвечающего потребностям пользователей и других заинтересованных сторон. Этот жизненный цикл охватывает от зарождения идей до вывода системы из эксплуатации. Он обеспечивает процессы приобретения и поставки систем. Это помогает улучшить взаимодействие и сотрудничество между сторонами, которые создают, используют и управляют современными программными системами, чтобы они могли работать интегрировано и согласованно. Кроме того, эта структура обеспечивает оценку и совершенствование процессов жизненного цикла.

Процессы в этом документе образуют исчерпывающий набор, из которого организация может построить модели жизненного цикла программного обеспечения, соответствующие ее продуктам и услугам. Организация, в зависимости от своей цели, может выбрать и применить соответствующее подмножество для достижения этой цели.

Этот документ можно использовать в одном или нескольких из следующих режимов:

a) Организацией ‒ чтобы помочь создать среду желаемых процессов. Эти процессы могут поддерживаться инфраструктурой методов, процедур, приемов, инструментов и обученного персонала. Затем организация может использовать эту среду для выполнения своих проектов и управления ими, а также для продвижения программных систем на этапах их жизненного цикла. В этом режиме данный документ используется для оценки соответствия заявленной, установленной среды его положениям.

b) Проектом ‒ чтобы помочь выбрать, структурировать и использовать элементы установленной среды для предоставления продуктов и услуг. В этом режиме данный документ используется при оценке соответствия проекта заявленной и установленной среде.

c) Покупателем и поставщиком ‒ чтобы помочь разработать соглашение относительно процессов и деятельности. Через соглашения процессы и виды деятельности, описанные в данном документе, выбираются, обсуждаются, согласовываются и выполняются. В этом режиме этот документ используется в качестве руководства при разработке соглашения.

d) Для специалистов по оценке процессов ‒ служить эталонной моделью процесса для использования при проведении оценки процессов, которая может быть использована для поддержки улучшения организационных процессов.

**ГОСУДАРСТВЕННЫЙ СТАНДАРТ РЕСПУБЛИКИ БЕЛАРУСЬ**

**СИСТЕМНАЯ И ПРОГРАММНАЯ ИНЖЕНЕРИЯ**

**ПРОЦЕССЫ ЖИЗНЕННОГО ЦИКЛА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ**

**CIСТЭМНАЯ I ПРАГРАМНАЯ IНЖЫНЕРЫЯ**

**ПРАЦЭСЫ ЖЫЦЦЕВАГА ЦЫКЛА ПРАГРАМНАГА ЗАБЯСПЯЧЭННЯ**

**Systems and software engineering - Software life cycle processes**

**Дата введения**

# 1 Сфера действий

1.1 Общие сведения

Этот документ устанавливает общую структуру для процессов жизненного цикла программного обеспечения с четко определенной терминологией, на которую может ссылаться индустрия программного обеспечения. Он содержит процессы, действия и задачи, которые применимы во время приобретения, поставки, разработки, эксплуатации, обслуживания или утилизации программных систем, продуктов и услуг. Эти процессы жизненного цикла достигаются за счет вовлечения заинтересованных сторон с конечной целью достижения удовлетворенности потребителей.

Этот документ применяется к приобретению, поставке, разработке, эксплуатации, техническому обслуживанию и утилизации (независимо от того, выполняются ли они внутри организации или за ее пределами) программных систем, продуктов и услуг, а также программной части любой системы. Программное обеспечение включает программную часть микропрограммного обеспечения. Включены те аспекты определения системы, которые необходимы для обеспечения особенностей программных продуктов и услуг.

Этот документ также предоставляет процессы, которые могут быть использованы для определения, контроля и улучшения процессов жизненного цикла программного обеспечения в организации или проекте.

Процессы, действия и задачи этого документа могут также применяться во время приобретения системы, содержащей программное обеспечение, отдельно или в сочетании с [ISO/IEC/IEEE 15288:2015].

В контексте данного документа и [ISO/IEC/IEEE 15288] существует целый ряд систем промышленного происхождения от систем, которые мало или вообще не используют программное обеспечение, до тех, в которых программное обеспечение представляет основной интерес. Редко можно встретить сложную систему без программного обеспечения. Все программные системы требуют физических компонентов системы (аппаратных средств) для работы, либо как часть интересующей программной системы, либо как вспомогательная система или инфраструктура. Таким образом, выбор применения данного документа для процессов жизненного цикла программного обеспечения или [ISO/IEC/IEEE 15288:2015], зависит от интересующей системы. Процессы в обоих документах имеют одинаковую цель процесса и результаты процесса, но отличаются деятельностью и задачами для выполнения программной инженерии или системной инженерии.

1.2 Цель

Цель документа ‒ предоставить определенный набор процессов для облегчения взаимодействия между покупателями, поставщиками и другими заинтересованными сторонами в жизненном цикле программной системы.

Данный документ написан для заказчиков, поставщиков, разработчиков, интеграторов, операторов, технического персонала, менеджеров, менеджеров по обеспечению качества и пользователей программных систем, продуктов и услуг. Он может использоваться одной организацией в самостоятельном режиме или в условиях с участием нескольких сторон. Стороны могут быть из одной организации или из разных организаций, а условия могут варьироваться от неформального соглашения до официального контракта.

Процессы, описанные в этом документе, могут использоваться в качестве основы для создания бизнес-среды, например, методов, процедур, методов, инструментов и обученного персонала. Приложение A содержит нормативные указания относительно настройки этих процессов жизненного цикла программного обеспечения.

1.3 Область применения

Документ применяется к полному жизненному циклу программных систем, продуктов и услуг, включая концепцию, разработку, производство, использование, поддержку и вывод из эксплуатации, а также к их приобретению и поставке, независимо от того, выполняются ли они внутри организации или за ее пределами. Процессы жизненного цикла этого документа могут применяться одновременно, итеративно и рекурсивно к программной системе и шаг за шагом к ее элементам.

Существует большое разнообразие программных систем с точки зрения их назначения, области применения, сложности, размера, новизны, адаптивности, количества, местоположения, периода эксплуатации и доработки. В этом документе описаны процессы, составляющие жизненный цикл созданных человеком программных систем. Следовательно, он применяется к уникальным программным системам, программным системам для широкого коммерческого или общедоступного распространения, а также настраиваемым, адаптируемым программным системам. Это также относится к полной автономной системе программного обеспечения и к программным системам, которые встроены и интегрированы в более крупные, более сложные и полные системы.

В этом документе представлена эталонная модель процесса, характеризуемая с точки зрения цели процесса и результатов процесса, которые являются результатом успешного выполнения задач деятельности. В Приложении B перечислены примеры артефактов и информационных элементов, которые могут быть связаны с различными процессами. Таким образом, этот документ можно использовать в качестве эталонной модели для поддержки оценки процесса, как указано в [ISO/IEC 33002: 2015]. В Приложении C представлена информация об использовании процессов жизненного цикла программного обеспечения в качестве эталонной модели процесса. Приложение D описывает конструкции процесса для использования в эталонной модели процесса. Приложение I обеспечивает соответствие между этим документом и [ISO/IEC/IEEE 12207: 2008] на уровне имени процесса и результата процесса.

1.4 Ограничения

Данный документ не предписывает конкретную модель жизненного цикла программного обеспечения, методологию разработки, метод, подход к моделированию или приемы работы. Пользователи этого документа несут ответственность за выбор модели жизненного цикла проекта и отображение процессов, действий и задач в этом документе внутри этой модели. Стороны также несут ответственность за выбор и применение соответствующих методологий, методов, моделей и приемов, подходящих для проекта.

Данный документ не устанавливает систему менеджмента и не требует использования какого-либо стандарта системы менеджмента. Однако он предназначен для совместимости с системой менеджмента качества, определенной стандартом ISO 9001, системой менеджмента услуг, определенной стандартом [ISO/IEC 20000-1 (IEEE Std 20000-1)], и системой менеджмента информационной безопасности, определенной стандартом [ISO/IEC 27000].

Данный документ не детализирует элементы информации с точки зрения имени, формата, явного содержания и носителей записи. [ISO/IEC/IEEE 15289] рассматривает содержание информационных единиц процесса жизненного цикла (документации).

# 2 Нормативные ссылки

В данном документе нет нормативных ссылок.

# 3 Термины, определения и сокращения

3.1 Термины и определения

Для целей настоящего документа применяются следующие термины и определения.

ISO и IEC ведут терминологические базы данных для использования в стандартизации по следующим адресам:

– IEC Электропедия: http://www.electropedia.org;

– ISO Платформа для онлайн-просмотра: http://www.iso.org/obp;

– IEEE Словарь стандартов онлайн: http://ieeexplore.ieee.org/xpls/dictionary.jsp.

Определения других терминов, как правило, можно найти в [ISO/IEC/IEEE 24765], Системная и программная инженерия ‒ Словарь, <www.computer.org/sevocab>.

**3.1.1** **приобретатель:** Заинтересованная сторона, которая приобретает или закупает продукт или услугу у поставщика

Примечание ‒ Другие термины, обычно используемые для обозначения приобретатель, – покупатель, клиент, владелец, закупщик или внутренний/организационный спонсор.

**3.1.2 приобретение:** Процесс получения системы, продукта или услуги.

**3.1.3 активность:** Набор взаимосвязанных задач процесса.

**3.1.4 гибкая разработка:** Подход к разработке программного обеспечения, основанный на последовательной разработке, частых проверках и адаптации, а также дополнительных поставках, при котором требования и решения развиваются в результате сотрудничества в межфункциональных группах и посредством постоянной обратной связи с заинтересованными сторонами. [ISO/IEC/IEEE 26515: 2011].

**3.1.5 соглашение:** Взаимное признание условий, на которых ведутся рабочие отношения.

***Пример ‒ Контракт, меморандум о соглашении.***

**3.1.6 архитектура:** <Система> фундаментальные концепции или свойства системы в ее среде, воплощенные в ее элементах, отношениях, а также в принципах ее проектирования и развития, [ISO/IEC/IEEE 42010:2011].

**3.1.7 базовая архитектура:** Соглашения, принципы и практики для описания архитектур, установленных в конкретной области применения и / или сообществе заинтересованных сторон.

***Примеры***

***1 Обобщенная эталонная архитектура и методологии предприятия (GERAM) [ISO 15704] –базовая архитектура.***

***2 Эталонная модель открытой распределенной обработки (RM – ODP) [ISO/IEC 10746] –базовая архитектура.***

ISO/IEC/IEEE 42010:2011].

**3.1.8 архитектурный вид:** Рабочий продукт, выражающий архитектуру системы с точки зрения конкретных системных проблем, ISO/IEC/IEEE 42010:2011].

**3.1.9 архитектурное представление:** Рабочий продукт, устанавливающий соглашения для построения, интерпретации и использования архитектурных видов для определения конкретных системных проблем, [ISO/IEC/IEEE 42010:2011].

**3.1.10 аудит:** Независимая проверка рабочего продукта или набора рабочих продуктов для оценки соответствия спецификациям, стандартам, договорным соглашениям или другим критериям.

**3.1.11 базовая линия:** Официально утвержденная версия элемента конфигурации, независимо от носителя, официально назначенная и фиксированная в определенное время в течение жизненного цикла элемента конфигурации, [IEEE Std 828‐2012].

**3.1.12 бизнес процесс:** Частично упорядоченный набор мероприятий предприятия, который может быть выполнен для достижения желаемого конечного результата в достижении заданной цели организации.

**3.1.13 концепция деятельности:** Устное и/или графическое изложение в общих чертах предположений или намерений организации в отношении действия или серии действий.

Примечания

1 Концепция деятельности часто воплощается в долгосрочных стратегических планах и годовых операционных планах. В последнем случае понятие деятельности в плане охватывает серию связанных действий, которые должны выполняться одновременно или последовательно. Концепция призвана дать общую картину деятельности организации. См. Также принцип работы (3.1.28).

2 Это обеспечивает основу для ограничения операционного пространства, возможностей системы, интерфейсов и операционной среды.

[ANSI/AIAA G‐043A‐2012e].

**3.1.14 касательство:** <Система> интерес к системе, имеющей отношение к одному или нескольким заинтересованным сторонам.

Примечание – Касательство относится к любому влиянию на систему в ее среде, включая развитие, технологическое, деловое, операционное, организационное, политическое, экономическое, правовое, нормативное, экологическое.

[ISO/IEC/IEEE 42010:2011].

**3.1.15 элемент конфигурации:** Элемент или совокупность оборудования, программного обеспечения или того и другого, который предназначен для управления конфигурацией и рассматривается как единый объект в процессе управления конфигурацией.

***Пример – Программное обеспечение, прошивка, данные, оборудование, люди, процессы (например, процессы для предоставления услуг пользователям), процедуры (например, инструкции оператора и руководства пользователя), средства, услуги, материалы и естественные объекты.***

**3.1.16 клиент:** Организация или лицо, получающее продукт или услугу.

***Пример – Потребитель, клиент, пользователь, заказчик, покупатель, или закупщик.***

Примечание – Приобретатель может быть внутренним или внешним по отношению к организации.

**3.1.17 проектировать:** <Процесс> для определения архитектуры, системных элементов, интерфейсов и других характеристик системы или системного элемента, [ISO/IEC/IEEE 24765:2010].

**3.1.18 проект:** Результат процесса в 3.1.17.

Примечания

1 Информация, включая спецификацию элементов системы и их взаимосвязей, достаточно полная для поддержки соответствующей реализации архитектуры.

2 Проект обеспечивает подробную физическую структуру на уровне реализации, поведение, временные отношения и другие атрибуты системных элементов.

**3.1.19 проектная характеристика:** Атрибуты проекта или отличительные особенности, которые относятся к измеримому описанию продукта или услуги.

**3.1.20 обеспечивающая система:** Система, которая поддерживает интересующую систему на этапах ее жизненного цикла, но не обязательно вносит непосредственный вклад в ее функционирование во время эксплуатации.

***Пример – Система управления конфигурацией, используемая для управления элементами программного обеспечения во время разработки программного обеспечения.***

Примечание –Каждая обеспечивающей (вспомогательная) система имеет собственный жизненный цикл. Этот документ применим к каждой обеспечивающей системе, когда она сама по себе рассматривается как интересующая система.

**3.1.21 среда:** <Система> контекст, определяющий обстановку и обстоятельства всех воздействий на систему, [ISO/IEC/IEEE 42010:2011].

**3.1.22 технические средства:** Физические средства или оборудование для облегчения выполнения какого-либо действия, например, здания, приборы, инструменты.

**3.1.23 инцидент:** Аномальное или неожиданное событие, набор событий, состояние или ситуация в любое время в течение жизненного цикла проекта, продукта, услуги или системы.

**3.1.24 элемент информации:** Отдельно идентифицируемая совокупность информации, которая производится, хранится и поставляется для использования человеком, [ISO/IEC/IEEE 15289:2015.

**3.1.25 инфраструктура:** Аппаратная и программная среда для поддержки проектирования, разработки и модификации компьютерных систем и программного обеспечения.

**3.1.26 жизненный цикл:** Эволюция системы, продукта, услуги, проекта или другого созданного человеком объекта от замысла до выхода из эксплуатации.

**3.1.27 модель жизненного цикла:** Система процессов и действий, связанных с жизненным циклом, которые могут быть организованы в этапы, выступающие в качестве общего ориентира для общения и понимания.

**3.1.28 операционная концепция:** Словесное и графическое изложение предположений или намерений организации в отношении операции или серии операций системы или связанного набора систем.

Примечание – Операционная концепция призвана дать общее представление об операциях с использованием одной или нескольких конкретных систем или набора связанных систем в операционной среде организации с точки зрения пользователей и операторов. См. также концепцию деятельности (3.1.13).

**3.1.29 оператор:** Физическое или юридическое лицо, выполняющее операции с системой.

Примечания

1 Роль оператора и роль пользователя могут быть возложены одновременно или последовательно на одно и то же лицо или организацию.

2 Отдельного оператора в сочетании со знаниями, навыками и процедурами можно рассматривать как элемент системы.

3 Оператор может выполнять операции в системе, которая работает, или в системе, которая работает, в зависимости от того, размещены ли рабочие инструкции в границах системы.

**3.1.30 организация:** Группа людей и объектов с распределением обязанностей, полномочий и взаимоотношений.

***Пример – Компания, корпорация, фирма, предприятие, учреждение, благотворительная организация, индивидуальный предприниматель, ассоциация, их части или комбинации.***

Примечание – Выявленная часть организации (даже такая небольшая, как отдельное лицо) или идентифицированная группа организаций могут рассматриваться как организация, если у нее есть обязанности, полномочия и взаимоотношения. Группа лиц, организованная с определенной целью, например клуб, союз, корпорация или общество, является организацией.

**3.1.31 сторона:** Организация, заключающая соглашение.

Примечание – В данном документе согласующие стороны называются приобретатель и поставщик.

**3.1.32 проблема:** Трудность, неопределенность или иным образом осознанное и нежелательное событие, совокупность событий, условие или ситуация, требующая расследования и корректирующих действий.

**3.1.33 процесс:** Набор взаимосвязанных или взаимодействующих действий, которые преобразуют входы в выходы.

**3.1.34 результат процесса:** Наблюдаемый результат успешного достижения цели процесса.

**3.1.35 цель процесса:** Цель высокого уровня выполнения процесса и вероятные результаты эффективного выполнения процесса.

Примечание – Целью внедрения процесса является предоставление выгод заинтересованным сторонам.

**3.1.36 продукт:** Результат процесса.

Примечание – Существует четыре согласованные общие категории продукции: оборудование (например, механическая часть двигателя); программное обеспечение (например, процедуры компьютерной программы и, возможно, соответствующая документация и данные); услуги (например, транспортировка); и обработанные материалы (например, смазочные материалы). Оборудование и обработанные материалы, как правило, являются материальными продуктами, в то время как программное обеспечение или услуги, как правило, являются нематериальными.

**3.1.37 проект:** Попытка с определенными критериями начала и завершения, предпринятая для создания продукта или услуги в соответствии с определенными ресурсами и требованиями.

Примечание – Проект иногда рассматривается как уникальный процесс, состоящий из скоординированных и контролируемых действий и включающий действия из процессов технического менеджмента и технических процессов, определенных в данном документе.

**3.1.38 <проект> проектный портфель:** Совокупность проектов, направленных на достижение стратегических целей организации.

**3.1.39 квалификация:** Процесс демонстрации того, способна ли организация выполнять установленные требования.

**3.1.40 контроль качества:** Часть управления качеством, направленная на обеспечение уверенности в том, что требования к качеству будут выполнены, [ISO 9000:2015].

**3.1.41 показатель качества:** Неотъемлемая характеристика продукта, процесса или системы, связанная с требованием.

Примечание – Критические характеристики качества обычно включают те, которые связаны со здоровьем, безопасностью, обеспечением безопасности, надежностью, доступностью и возможностью поддержки.

**3.1.42 менеджмент качества:** Скоординированная деятельность по руководству и контролю организации в отношении качества.

**3.1.43 релиз:** Конкретная версия элемента конфигурации, предоставляемая для определенной цели.

***Пример – Тест релиз.***

**3.1.44 требование:** Утверждение, которое объясняет или выражает потребность и связанные с ней ограничения и условия, [ISO/IEC/IEEE 29148:2011].

**3.1.45 ресурс:** Актив, который используется или потребляется во время выполнения процесса.

Примечание – Ресурсы включают в себя повторно используемые, возобновляемые или расходные материалы.

***Пример – различные объекты, такие как финансирование, персонал, помещения, основное оборудование, инструменты и коммунальные услуги, такие как электроэнергия, вода, топливо и коммуникационная инфраструктур.***

**3.1.46 вывод из эксплуатации:** Прекращение активной поддержки со стороны организации по эксплуатации и техническому обслуживанию, частичная или полная замена на новую систему или установка модернизированной системы.

**3.1.47 риск:** Влияние неопределенности на цели.

Примечания

1 Эффект – это отклонение от ожидаемого – положительное или отрицательное. Положительный эффект также известен как возможность.

2 Цели могут иметь различные аспекты (например, финансовые цели, цели в области здравоохранения и безопасности, экологические цели) и могут применяться на различных уровнях (например, стратегическом, общеорганизационном, проектном, продуктовом и процессном).

3 Риск часто характеризуется ссылкой на потенциальные события и последствия, или их сочетание.

4 Риск часто выражается в виде комбинации последствий события (включая изменения в обстоятельствах) и связанной с ним вероятности возникновения.

5 Неопределенность – это состояние, даже частичное, недостатка информации, относящейся к пониманию или знанию события, его последствий или вероятности.

[ISO Guide 73:2009, определение 1.1].

**3.1.48 надежность:** Ожидание того, что при определенных условиях система не приведет к состоянию, при котором жизнь, здоровье, имущество или окружающая среда человека окажутся под угрозой.

**3.1.49 безопасность:** Защита от преднамеренной диверсии или принудительного отказа; совокупность четырех атрибутов – конфиденциальность, целостность, доступность и подотчетность – плюс аспекты пятого, удобство использования, все из которых связаны с вопросом их обеспечения, [NATO AEP‐67].

**3.1.50 услуга:** Выполнение деятельности, работы или обязанностей.

Примечание

1 Услуга является самодостаточной, последовательной, дискретной и может быть составлена из других услуг.

2 Услуга, как правило, является нематериальным продуктом.

**3.1.51 программный элемент:** Элемент системы, который является программным обеспечением.

**3.1.52 проектирование программного обеспечения:** Применение систематического, дисциплинированного, поддающегося количественной оценке подхода к разработке, эксплуатации и сопровождению программного обеспечения; то есть применение инженерии к программному обеспечению.

**3.1.53 программный элемент:** Исходный код, объектный код, управляющий код, управляющие данные или совокупность этих элементов.

Примечание – Программный элемент можно рассматривать как системный элемент данного документа и стандарта [ISO/IEC 15288:2015. Программные элементы обычно являются элементами конфигурации].

**3.1.54 программный продукт:** Набор компьютерных программ, процедур и, возможно, связанной с ними документации и данных.

Примечание – Программный продукт – это программная система, рассматриваемая как выход (продукт), полученный в результате процесса.

**3.1.55 программная система:** Система, для которой программное обеспечение имеет первостепенное значение для заинтересованных сторон.

Примечания

1 В наиболее общем случае программная система состоит из оборудования, программного обеспечения, людей и ручных процедур.

2 В программной системе программное обеспечение является ведущей движущей силой в удовлетворении системных требований.

**3.1.56 элемент программной системы:** Член набора элементов, составляющих программную систему

Примечания

1 Элемент программной системы может включать один или несколько программных блоков, программных элементов, аппаратных блоков, аппаратных элементов, служб и других системных элементов и систем.

2 Элемент программной системы можно рассматривать как элемент системы.

**3.1.57 программный блок:** Программный компонент атомарного уровня в архитектуре программного обеспечения, который может быть подвергнут автономному тестированию.

Примечание – Некоторые программные блоки представляют собой отдельные компилируемые фрагменты кода.

[ISO 26262‐1:2011].

**3.1.58 этап:** Период в жизненном цикле объекта, относящийся к состоянию его описания или реализации.

Примечания

1 В данном документе этапы относятся к основным этапам прогресса и достижений организации на протяжении ее жизненного цикла.

2 Этапы часто пересекаются.

**3.1.59 заинтересованная сторона:** Лицо или организация, имеющие право, долю, претензии или интерес в системе или в обладании ею характеристиками, которые отвечают их потребностям и ожиданиям.

***Пример – Конечные пользователи, организации конечных пользователей, сторонники, разработчики, производители, инструкторы, специалисты по обслуживанию, утилизации, покупатели, организации-поставщики и регулирующие органы.***

Примечание – У некоторых вовлеченных сторон могут быть интересы, противоположные друг другу или противостоящие системе.

**3.1.60 поставщик:** Организация или физическое лицо, заключающее с приобретателем договор на поставку товара или услуги.

Примечания

1 Другие термины, обычно используемые для обозначения поставщика: подрядчик, производитель, продавец или разработчик.

2 Приобретатель и поставщик иногда являются частью одной и той же организации.

**3.1.61 система:** Сочетание взаимодействующих элементов, организованных для достижения одной или нескольких заявленных целей.

Примечания

1 Систему иногда рассматривают как продукт или как услуги, которые она предоставляет.

2 На практике толкование его значения часто уточняется за счет использования ассоциативного существительного, например, авиационная система или система управления базой данных. В качестве альтернативы слово "система" заменяется просто синонимом, зависящим от контекста, например, самолет или база данных, хотя это потенциально скрывает перспективу системных принципов.

3 Система может включать в себя соответствующее оборудование, средства, материалы, программное обеспечение, встроенное ПО, техническую документацию, услуги и персонал, необходимые для работы и поддержки, в той степени, которая необходима для использования в предполагаемой среде.

4 См. Для сравнения: обеспечивающая система, рассматриваемая система, система систем.

**3.1.62 элемент системы:** Член набора элементов, составляющих систему.

***Пример – Аппаратные средства, программное обеспечение, данные, люди, процессы (например, процессы предоставления услуг пользователям), процедуры (например, инструкции оператора), объекты, материалы и естественно возникающие сущности или любая их комбинация.***

Примечание – Элемент системы – это дискретная часть системы, которая может быть реализована для выполнения заданных требований.

**3.1.63 рассматриваемая система SOI:** Система, жизненный цикл которой рассматривается.

**3.1.64 суперсистема SoS:** Набор систем, которые интегрируются или взаимодействуют для обеспечения уникальных возможностей, которые ни одна из составляющих систем не может реализовать самостоятельно.

Примечание – Каждая составная система является полезной системой сама по себе, имеет собственное управление, цели и ресурсы, но координируется в рамках SoS для обеспечения уникальных возможностей SoS.

**3.1.65 системное проектирование:** Междисциплинарный подход, регулирующий общие технические и управленческие усилия, необходимые для преобразования набора потребностей, ожиданий и ограничений заинтересованных сторон в решение и для поддержки этого решения на протяжении всего срока его службы.

**3.1.66 задача:** Требуемое, рекомендуемое или допустимое действие, призванное способствовать достижению одного или нескольких результатов процесса.

**3.1.67 техническое руководство:** Применение технических и административных ресурсов для планирования, организации и контроля инженерных функций.

**3.1.68 компромисс:** Действия по принятию решений, которые выбирают из различных требований и альтернативных решений на основе чистой выгоды для заинтересованных сторон.

**3.1.69 трассируемость (отслеживаемость):** Степень, в которой отношения могут быть установлены между двумя или более логическими объектами, особенно объектами, имеющими отношения предшественник-преемник или главный-подчиненный друг к другу, такие как требования, системные элементы, проверки или задачи.

***Пример – Программные функции и тестовые примеры обычно соотносятся с требованиями к программному обеспечению.***

**3.1.70 пользователь:** Человек или группа, которые взаимодействуют с системой или извлекают выгоду из системы во время ее использования.

Примечание – Роль пользователя и роль оператора иногда возлагается, одновременно или последовательно, на одного и того же человека или организацию.

[ISO/IEC 25010:2011].

**3.1.71 валидация:** Подтверждение, посредством предоставления объективных доказательств, того, что требования для конкретного предполагаемого использования или применения были выполнены.

Примечания

1 Система способна выполнять предполагаемое использование, цели и задачи (т. Е. Удовлетворять требования заинтересованных сторон) в предполагаемой операционной среде. Была построена правильная система.

2 В контексте жизненного цикла валидация включает в себя набор действий для получения уверенности в том, что система способна выполнить свое предполагаемое использование, цели и задачи в среде, подобной операционной среде.

**3.1.72 верификация:** Подтверждение, посредством предоставления объективных доказательств, того, что установленные требования были выполнены.

Примечание – Верификация – это набор действий, которые сравнивают систему или элемент системы с требуемыми характеристиками. Это включает в себя, но не ограничивается заданными требованиями, дизайном, описаниями и самой системой. Система была построена правильно.

[ISO 9000:2015]/

3.2 Сокращения

CCB – Configuration Control Board – комиссия по контролю за конфигурацией;

CM – Configuration Management – конфигурационное управление;

COTS – Commercial‐Off‐The‐Shelf – готовое коммерческое изделие без доработки;

FCA – Functional Configuration Audit – функциональная проверка конфигурации;

FOSS – Free and Open Source Software – Программное обеспечение с открытыми исходными кодами;

GUI – Graphical User Interface – графический интерфейс пользователя;

NDI – Non‐Developmental Items – Элементы, не связанные с развитием;

QA – Quality Assurance – обеспечение качества;

PCA – Physical Configuration Audit – аудит физической конфигурации;

PESTEL – Political, Economic, Social, Technological, Environmental, and Legal – Политические, экономические, социальные, технологические, экологические и правовые аспекты;

PMI – Project Management Institute – институт по управлению проектами;

PMP – Project Management Plan – план управления проектом;

PRM – Process Reference Model – базовая модель процесса;

SCM – Software Configuration Management – управление составом программного обеспечения;

SDP – Software Development Plan – план разработки программного обеспечения;

SEMP – Systems Engineering Management Plan – план управления системной инженерией;

SOI – System‐of‐Interest – рассматриваемая система;

SoS – System of Systems – суперсистема;

SWOT – Strengths, Weaknesses, Opportunities, Threats – сильные стороны, слабые стороны, возможности, угрозы;

WBS – Work Breakdown Structure – структура разделения работ.

# 4 Соответствие

4.1 Предполагаемое использование

Требования данного документа содержатся в Пункте 6 и Приложении А. Данный документ устанавливает требования для ряда процессов, подходящих для использования в течение жизненного цикла программной системы или продукта. Признано, что конкретные проекты или организации могут не нуждаться в использовании всех процессов, предусмотренных данным документом. Поэтому реализация данного документа обычно включает в себя выбор и декларирование набора процессов, подходящих для организации или проекта. Существует два способа утверждать, что реализация соответствует положениям данного документа ‒ полное соответствие и адаптированное соответствие.

Есть два критерия для утверждения о полном соответствии. Достижение любого критерия достаточно для соответствия, хотя выбранный критерий (или критерии) должен быть указан в заявлении. Утверждение «полное соответствие задачам» означает, что все требования к действиям и задачам заявленного набора процессов выполнены. В качестве альтернативы утверждение «полного соответствия результатам» означает, что все требуемые результаты заявленного набора процессов достигнуты.

Полное соответствие результатам обеспечивает большую свободу в реализации соответствующих процессов и может быть полезно для реализации процессов, которые будут использоваться в контексте инновационной модели жизненного цикла.

Примечания

1 Варианты соответствия предусмотрены для обеспечения необходимой гибкости при применении данного документа. Каждый процесс имеет набор целей (сформулированных как "результаты") и набор видов деятельности и задач, которые представляют собой один из способов достижения целей.

2 Пользователи, реализующие деятельность и задачи заявленного набора процессов, могут утверждать о полном соответствии задачам выбранных процессов. Некоторые пользователи, однако, могут иметь инновационные варианты процессов, которые достигают целей (т.е. результатов) заявленного набора процессов без реализации всех видов деятельности и задач. Такие пользователи могут заявить о полном соответствии результатам заявленного набора процессов. Эти два критерия - соответствие задаче и соответствие результату - обязательно не эквивалентны, поскольку конкретное выполнение действий и задач может требовать в некоторых случаях более высокого уровня возможностей, чем просто достижение результатов.

3 Если данный документ используется для разработки соглашения между приобретателем и поставщиком, пункты этого документа могут быть выбраны для включения в соглашение с изменениями или без них. В этом случае приобретателю и поставщику целесообразнее заявить о соответствии соглашению, чем о соответствии данному документу.

4 Организация (например, национальная, промышленная ассоциация, компания), навязывающая этот документ, в качестве условия торговли может определить и опубликовать минимальный набор требуемых процессов, результатов, действий и задач, которые определяют соблюдение поставщиками условий. торговли.

5Требования этого документа отмечены глаголом «должен». Рекомендации помечаются глаголом «следует». Разрешения отмечаются глаголом «может». Однако, несмотря на глагол, который используется, требования для соответствия выбираются, как описано ранее.

4.2 Полное соответствие

4.2.1 Полное соответствие результатам

Заявление о полном соответствии декларирует набор процессов, для которых заявлено соответствие. Полное соответствие результатам достигается путем демонстрации того, что все результаты заявленного набора процессов были достигнуты. В этой ситуации положения для действий и задач заявленного набора процессов являются скорее руководством, чем требованиями, независимо от глагольной формы, которая используется в положении.

Одно из предполагаемых применений этого документа ‒ облегчить оценку и улучшение процесса. Для этого цели каждого процесса записываются в форме «результатов», совместимых с положениями [ISO/IEC33002]. Этот стандарт предусматривает оценку процессов, описанных в этом документе, обеспечивая основу для улучшений. Пользователи, намеревающиеся оценить и улучшить процесс, могут использовать результаты процесса, описанные в этом документе, в качестве «эталонной модели процесса», требуемой [ISO/IEC 33002].

4.2.2 Полное соответствие задачам

Заявление о полном соответствии декларирует набор процессов, для которых заявлено соответствие. Полное соответствие задачам достигается путем демонстрации того, что все требования действий и задач заявленного набора процессов были выполнены. В этой ситуации положения о результатах заявленного набора процессов являются скорее руководством, чем требованиями, независимо от глагольной формы, используемой в положении.

Примечание ‒ Заявление о полном соответствии задачам может быть уместным в контрактных ситуациях, когда покупатель или регулирующий орган требует детального понимания процессов поставщиков.

4.2.3 Индивидуальное соответствие

Когда этот документ используется в качестве основы для создания набора процессов, которые не соответствуют требованиям для полного соответствия, разделы этого документа выбираются или изменяются в соответствии с процессом адаптации, предписанным в Приложении A. Адаптированный текст, для которого адаптировано соответствие, востребовано, заявлено. Индивидуальное соответствие достигается за счет демонстрации того, что результаты, действия и задачи в том виде, в каком они были адаптированы, были достигнуты.

# 5 Ключевые концепции и применение

5.1 Введение

Этот раздел включен, чтобы выделить и помочь объяснить основные концепции, на которых основан этот документ.

Примечание – Дальнейшее развитие данных концепций можно найти в [ISO/IEC TS 24748-1, ISO/IEC TR 24748-2 и ISO/IEC TR24748-3], посвященных применению менеджмента жизненного цикла.

5.2 Концепции программных систем

5.2.1 Программные системы

Программные системы, рассматриваемые в данном документе, созданы и используются для предоставления продуктов или услуг в определенных средах на благо пользователей и других заинтересованных сторон. Эти программные системы могут включать в себя следующие системные элементы: аппаратное обеспечение, программное обеспечение, данные, людей, процессы (например, процессы для предоставления услуг пользователям), процедуры (например, инструкции оператора), помещения, услуги, материалы и естественные объекты. С точки зрения пользователя они рассматриваются как продукты или услуги.

Данный документ применяется к системам, для которых программное обеспечение имеет первостепенное значение для заинтересованных сторон. Он основан на общих принципах системной инженерии и разработки программного обеспечения. Фундаментальная предпосылка этого документа заключается в том, что программное обеспечение всегда существует в контексте системы. Поскольку программное обеспечение не работает без оборудования, процессор, на котором выполняется программное обеспечение, можно рассматривать как часть системы. В качестве альтернативы, оборудование или услуги, на которых размещается программная система и обрабатывается связь с другими системами, также могут рассматриваться как обеспечивающие системы или внешние системы в операционной среде.

Восприятие и определение конкретной программной системы, ее архитектуры и ее элементов зависят от интересов и ответственности заинтересованных сторон. Система интересов одной заинтересованной стороны может рассматриваться как системный элемент в системе интересов другой заинтересованной стороны. Более того, интересующая система может рассматриваться как часть среды для интересующей системы другой заинтересованной стороны.

Ниже приведены ключевые моменты, касающиеся характеристик рассматриваемых систем:

a) определенные границы могут заключать в себе значимые потребности и практические решения;

b) возможность применения иерархической или другой взаимосвязи между элементами системы;

c) модуль на любом уровне в рассматриваемой системе может рассматриваться как система;

d) система может включать в себя интегрированный, определенный набор подчиненных системных элементов;

e) человек может рассматриваться как внешний по отношению к системе пользователь и как элемент системы (т.е. оператор) внутри системы; и

f) система может рассматриваться изолированно, как единое целое, т.е. продукт; или как набор функций, способных взаимодействовать с окружающей средой, т.е. набор услуг.

Независимо от границ, выбранных для определения системы, концепции в этом документе являются общими и позволяют практикующему специалисту соотносить или адаптировать отдельные экземпляры жизненных циклов к принципам своей системы.

5.2.2 Структура программной системы

В этом документе процессы жизненного цикла описаны применительно к программной системе, которая состоит из набора взаимодействующих элементов системы (включая элементы программного обеспечения), каждый из которых может быть реализован для выполнения соответствующих заданных требований (Рисунок 1). Таким образом, ответственность за внедрение любого элемента системы может быть делегирована другой стороне по соглашению.



**Рисунок 1 – Взаимосвязь программной системы и элемента программной системы**

Отношения между программной системой и ее полным набором системных элементов обычно можно представить, показывая отношения между элементами ‒ часто изображаемые в виде иерархии для самых простых рассматриваемых систем. Один их подходов к некоторым видам программной деятельности – декомпозиция. Другие подходы включают объектно-ориентированный подход, где элементы системы представлены в плоском (неиерархическом) описании, например, в сетевой диаграмме. Для более сложных программных рассматриваемых систем, может потребоваться рассмотреть предполагаемый элемент системы как систему (которая, в свою очередь, состоит из элементов системы), прежде чем можно будет с уверенностью определить полный набор элементов системы (Рисунок 2). Таким образом, соответствующие процессы жизненного цикла системы рекурсивно применяются к рассматриваемой системе для определения ее структуры до того момента, когда понятные и управляемые элементы программной системы могут быть реализованы (созданы, адаптированы, приобретены или повторно использованы).



**Рисунок 2 – Пример структуры рассматриваемой системы**

Хотя рисунки 1 и 2 подразумевают иерархические отношения, в действительности существует все большее количество систем, которые с одного или нескольких аспектов не являются иерархическими, например, сети и другие распределенные системы. В Приложении G рассматривается концепция суперсистема (SoS).

Примечание – Декомпозиция –фундаментальное действие для многих программных действий. Не все декомпозиции подразумевают обозначение новых элементов программной системы и соответствующее рекурсивное применение действия. Обозначение декомпозированной конструкции в качестве элемента необходимо только тогда, когда к ее разработке целесообразно применять отдельные требования, проектирование или действия по реализации. Одним из примеров подходящей ситуации является ситуация, когда элемент должен быть разработан отдельной организацией. Другой пример ‒ когда руководство определяет, что уместно четко отслеживать статус разработки или настройки элемента.

5.2.3 Обеспечивающие системы

На протяжении жизненного цикла рассматриваемой системы требуются важные услуги от систем, которые не являются непосредственно частью операционной среды рассматриваемой системы, например, система моделирования, система обучения, система технического обслуживания. Каждая из этих систем позволяет проводить часть, например, стадию жизненного цикла рассматриваемой системы. Называемые «вспомогательными системами», они способствуют продвижению рассматриваемой системы на протяжении ее жизненного цикла.

Взаимосвязь между услугами, предоставляемыми операционной среде рассматриваемой системы, и услугами, предоставляемыми вспомогательными системами рассматриваемой системе, показана на Рисунке 3. Можно увидеть, что вспомогательные системы косвенно способствуют оказанию услуг по рассматриваемой системе. Взаимосвязи между рассматриваемой системой и вспомогательными системами могут быть двунаправленными или односторонними. Помимо взаимодействия с вспомогательными системами, рассматриваемой система также может взаимодействовать с другими системами в операционной среде, показанными как Системы A, B и C. Включены требования к интерфейсам с вспомогательными системами и другими системами в операционной среде в требованиях к рассматриваемой системе.



**Рисунок 3 –Рассматриваемая программная система, ее операционная среда и вспомогательные системы**

На этапе жизненного цикла программного обеспечения соответствующие вспомогательные системы и рассматриваемая система рассматриваются вместе. Поскольку они взаимозависимы, их также можно рассматривать как систему. Когда подходящая вспомогательная система еще не существует, проект, который отвечает за рассматриваемую систему, может нести прямую ответственность за создание и использование вспомогательной системы. Создание вспомогательной системы можно рассматривать как отдельный проект, а впоследствии как еще одну рассматриваемую систему.

Дальнейшее развитие концепций представлено в [ISO/IEC/IEEE 24748 (все части)], описывающим применению процессов жизненного цикла.

Примечание – К обеспечивающим системам при разработке программного обеспечения относятся среды разработки и тестирования программного обеспечения для целевых платформ.

5.2.4 Процессы жизненного цикла для программной системы

В программной системе требования, архитектура и процессы проектирования на системном уровне приводят к распределению системных требований по различным элементам. Рассматриваемая программная система реализуется, в первую очередь, путем анализа требований к программной системе, архитектуры и дизайна и определения того, какие функции будут реализованы в программном обеспечении или других элементах, реализации программного обеспечения и других элементов и интеграции этих элементов в систему программного обеспечения. Следовательно, программный продукт или услугу можно рассматривать как элемент программной системы.

В некоторых случаях архитектурное определение программной системы может указывать на то, что уместно рассматривать ее как состоящую из набора отдельных подчиненных элементов. В свою очередь, каждый из элементов программного обеспечения можно рассматривать как отдельную программную систему, как описано ранее. В этих случаях этот документ может применяться в обратном порядке для получения или разработки подчиненных элементов.

Данный документ взаимосвязан с [ISO/IEC/IEEE 15288:2015] и в большей степени применим к программным системам. Для учета ситуаций, в которых применяются как [ISO/IEC/IEEE 15288:2015], так и [ISO/IEC/IEEE 12207:2017] (например, разработка системы, содержащей программное обеспечение, или разработка программной системы, содержащей аппаратное обеспечение), структуры их процессов согласованы и идентичны. Процессы данного документа напрямую соответствуют процессам [ISO/IEC/IEEE 15288] со специализацией для программных продуктов и услуг.

В случае, когда не программные элементы системы имеют первостепенное значение, организация может принять решение о применении [ISO/IEC/IEEE 15288] для выполнения соответствующих процессов, действий и задач жизненного цикла. Для каждого программного элемента системы организация может применять данный документ для создания, адаптации, приобретения или повторного использования программных элементов.

5.3 Организация и концепции проекта

5.3.1 Организации

Когда организация, в целом или как часть, заключает соглашение, ее иногда называют «стороной» соглашения. Стороны могут быть из одной организации или из разных организаций. Организация может представлять из себя отдельное лицо, если на него возложены обязанности и полномочия.

В неформальных терминах организация, ответственная за выполнение процесса, иногда называется именем этого процесса. Например, организацию, выполняющую процесс приобретения, иногда называют "приобретатель". Другие примеры включают поставщика, исполнителя, сопровождающей стороны и оператора.

В данном документе к организациям применяется несколько других терминов: «пользователь» может быть организацией или лицами, которые непосредственно участвуют или получают выгоду от использования продукта или услуги; "клиент" относится к пользователю и приобретателю вместе; и «заинтересованная сторона» относится к лицу или организации, имеющим интерес к системе.

Процессы и организации связаны только функционально. Данный документ не диктует и не подразумевает структуру организации, а также не указывает, что конкретные процессы должны выполняться конкретными частями организации. Организация, которая внедряет этот документ, несет ответственность за определение подходящей структуры организации и назначение соответствующих ролей для выполнения процессов.

Процессы, приведенные в этом документе, образуют всеобъемлющий набор для обслуживания различных организаций. Организация, малая или большая, в зависимости от своей деловой цели или стратегии приобретения, может выбрать соответствующий набор процессов (и связанных с ними действий и задач) для выполнения этой цели. Организация может выполнять один процесс или несколько процессов.

Данный документ предназначен для внутреннего или внешнего применения двумя или более организациями. При внутреннем применении две договаривающиеся стороны обычно действуют в соответствии с условиями соглашения, которое может различаться по формальности при различных обстоятельствах. При внешнем применении две договаривающиеся стороны обычно действуют в соответствии с условиями контракта. В этом документе термин «соглашение» используется в любой ситуации.

Для целей этого документа предполагается, что любой проект осуществляется в рамках организации. Важно, потому что проект зависит от различных результатов, производимых бизнес-процессами организации, например, количество сотрудников для укомплектования проекта и помещения для размещения проекта. Для данной цели в документе представлен набор процессов «Организационного обеспечения проекта». Предполагается, что эти процессы не являются достаточными для ведения бизнеса; вместо этого процессы, рассматриваемые как совокупность, предназначены для определения минимального набора зависимостей, которые проект накладывает на организацию.

5.3.2 Внедрение на уровне организации и проекта

Современные предприятия стремятся разработать надежный набор процессов жизненного цикла, которые многократно применяются к проектам и услугам бизнеса. Таким образом, этот документ предназначен для использования на уровне организации или проекта. Организация может принять документ и дополнить его соответствующими процедурами, практиками, инструментами и политиками. В свою очередь, проект организации обычно соответствует процессам организации, а не непосредственно этому документу.

В некоторых случаях проекты могут выполняться организацией, у которой нет соответствующего набора процессов, принятых на организационном уровне. Такой проект может применять положения данного документа непосредственно к проекту.

5.4 Концепции жизненного цикла

5.4.1 Этапы жизненного цикла программного обеспечения

Жизненные циклы различаются в зависимости от характера, назначения, использования и преобладающих условий программной системы. Использование стадий одновременно и в разном порядке может привести к формам жизненного цикла с совершенно разными характеристиками. Каждый этап имеет определенную цель и вклад в планирование и выполнение всего жизненного цикла программной системы. Согласно [ISO/IEC TS 24748-1], типичные этапы жизненного цикла системы включают концепцию, разработку, производство, использование, поддержку и вывод из эксплуатации. Использование этих терминов для определения этапов не является нормативным. Обычный набор этапов для программной системы - это исследование концепции, разработка, поддержка и вывод из эксплуатации, с переходами между этапами для системы в целом и для ее элементов.

Этапы представляют собой основные периоды жизненного цикла, связанные с программной системой, и они относятся к состоянию описания программной системы или самой программной системы. Этапы описывают основной прогресс и вехи достижений программной системы на протяжении ее жизненного цикла. Они дают начало основным этапам жизненного цикла для принятия решений. Эти этапы принятия решений используются организациями для понимания и управления неотъемлемыми неопределенностями и рисками, связанными с затратами, графиком и функциональностью при создании или использовании программной системы. Таким образом, использование этапов предоставляет организациям структуру, в рамках которой руководство организации имеет высокий уровень видимости и контроля над проектами и техническими процессами. Организации определяют и используют этапы по-разному, чтобы удовлетворить противоположные бизнес-стратегии и стратегии снижения рисков.

Процессы жизненного цикла, определенные в этом документе, не привязаны к какому-либо конкретному этапу жизненного цикла программного обеспечения. Все процессы жизненного цикла включают планирование, выполнение и оценку деятельности, которые следует рассматривать для использования на каждом этапе.

Дальнейшее развитие данных концепций представлено в [ISO/IEC/IEEE 24748 (все части)], описывающим применение управления жизненным циклом.

5.4.2 Модель жизненного цикла программной системы

У каждой программной системы есть жизненный цикл. Жизненный цикл можно описать с помощью абстрактной функциональной модели, которая представляет концептуальную концепцию потребности в системе, ее реализации, использования, развития и утилизации.

Программная система проходит через свой жизненный цикл как результат действий, выполняемых и управляемых работниками в организациях, использующих процессы для выполнения этих действий. Детализация в модели жизненного цикла выражается в терминах этих процессов, их результатов, отношений и последовательности.

Данный документ не предписывает какую-либо конкретную модель жизненного цикла. Вместо этого он определяет набор процессов, называемых процессами жизненного цикла, которые можно использовать при определении жизненного цикла системы. Кроме того, этот документ не предписывает какую-либо конкретную последовательность процессов в рамках модели жизненного цикла. Последовательность процессов определяется целями проекта и выбором модели жизненного цикла. Часто этап развития подразделяется более точно и различным образом.

Один из часто упоминаемых наборов этапов разработки программного обеспечения – выявление, требования, проектирование, построение и тестирование прогнозирующей или «каскадной» модели. Если этапы считаются последовательными, то каждый этап должен давать правильные результаты перед переходом к следующему этапу. На практике этого чрезвычайно сложно достичь, если хорошо не известны требования, а первоначальная смета затрат не точна. Выполняя этапы, существует риск выполнить обширную переделку, которая должным образом не попадает ни в один из запланированных этапов и, следовательно, вероятно, не попадает в рамки какого-либо бюджета.

Примечание – Уинстон Ройс, широко известный как один из первых аналитиков моделей процессов жизненного цикла, описал необходимость стадий доработки, а не «каскада» (термин, который он не использовал). К сожалению, этапы доработки были исключены из «каскадной» модели, как ее обычно понимали.

Для решения проблем неполного знания требований и неточных оценок, был предложен ряд других типов моделей: поэтапного приближения, спиральные, итерационные и эволюционные (адаптивные). Эти модели жизненного цикла могут включать гибкие техники и методы. Эти модели, как правило, могут включать повторяющееся выполнение процессов и стадий жизненного цикла в течение жизненного цикла, например, для различных этапов разработки программного продукта, для более точной обработки исключений из общих функций или для требований, которые не были полностью определены с самого начала. Данные модели можно применять на разных этапах, таких как разработка и использование или развертывание. Использование этих моделей может повлиять на стратегии выпуска программного обеспечения и стратегии приобретения программных сервисов.

***Пример – Элементы программного обеспечения могут разрабатываться поэтапно, а затем храниться для блочного оперативного выпуска в удобное время в бизнес-цикле организации.***

Модель «поэтапной разработки» включает в себя начальное планирование, первоначальный анализ требований, первоначальное определение архитектуры и первоначальную валидацию, но распределяет деятельность по проектированию, внедрению, проверке (а иногда и доставке) по сериям этапов, каждый из которых обеспечивает часть запланированного функционирования. Подход обеспечивает некоторую гибкость, позволяющую реагировать на неточные оценки затрат или графика, перемещая функциональные возможности на более поздние этапы.

«Спиральный» вариант поэтапного развития предлагает упорядочить разработку функциональности на основе риска, с учетом самых рискованных проблем на ранних стадиях. Это обеспечивает некоторую защиту от неожиданных затрат, возникающих на поздних этапах цикла разработки.

Модель «итеративной разработки» выполняет начальное планирование, а затем состоит из циклического процесса создания прототипа, тестирования, анализа и уточнения требований и решения. «Итерационные» модели многократно выполняют процессы жизненного цикла для более быстрого предоставления приоритетных системных функций, при этом уточненные или более сложные элементы системы появляются в более поздних итерациях.

«Эволюционная модель» предназначена для работы с неполным знанием требований. Она предусматривает начальное планирование и начальное определение архитектуры, но распределяет анализ требований, проектирование, создание, верификацию, валидацию и поставку на ряд этапов. Предоставленные возможности, которые не отвечают потребностям пользователей, могут быть доработаны на последующих этапах эволюции.

«Гибкие» методы можно применять в самых разных моделях. Хотя гибкие методы широко используются при выполнении эволюционной модели жизненного цикла, их можно использовать в других моделях жизненного цикла на различных этапах. Общим для этих методов является упор на непрерывный контроль и сотрудничество в быстром производстве работающего программного обеспечения в среде, где ожидаются изменения, в том числе изменения требований. В Приложении H представлена информация о применении этого документа в контексте гибкой разработки.

Примечания

1 Выбор названия типа модели не удовлетворяет требованию определения модели, состоящей из этапов с определенной целью и результатами, достигаемыми с помощью процессов, описанных в этом документе.

3 [ISO/IEC TS 24748-1, ISO/IEC TR 24748-2, ISO/IEC TR 24748-3 и ISO/IEC/IEEE 24748-4] содержат дополнительные подробности относительно моделей и стадий жизненного цикла. Модели, описанные в данном пункте, применимы не только к программным системам, но и к другим системам, как описано в [ISO/IEC/IEEE 15288:2015].

5.5 Концепции процесса

5.5.1 Критерии процессов

Определение процессов жизненного цикла в данном документе основано на трех основных принципах:

1) Каждый процесс жизненного цикла имеет тесные взаимосвязи между своими результатами, действиями и задачами.

2) Зависимости между процессами уменьшены до максимально возможной степени.

3) Процесс способен выполняться одной организацией в течение жизненного цикла.

5.5.2 Описание процессов

Каждый процесс данного документа описывается в терминах следующих атрибутов:

a) Название передает объем процесса в целом.

b) Цель описывает цели выполнения процесса.

c) Итоги выражают наблюдаемые результаты, ожидаемые от успешного выполнения процесса.

d) Деятельность представляет собой набор взаимосвязанных задач процесса.

e) Задачи реализуют требования, рекомендации или допустимые действия, направленные на поддержку достижения конечных результатов.

Процессы и группы процессов в данном документе идентичны по своим целям и результатам с процессами в [ISO/IEC/IEEE 15288:2015] за одним исключением: процесс «Определение требований к системе/программному обеспечению» в данном документе переименован из процесса «Определение системных требований» в [ISO/IEC/IEEE 15288:2015]. Чтобы подчеркнуть гармонизацию системных и программных системных процессов, в пункте 6 цели и результаты процесса представлены в виде врезок.

Действия, задачи и рабочие продукты, относящиеся к программному обеспечению, применяются для достижения результатов процессов, описанных в этом документе. В Приложении E представлены дополнительные представления о процессе.

Дополнительные сведения об форме описания процесса можно найти в [ISO/IEC TR 24774].

5.5.3 Общие характеристики процессов

В дополнение к основным атрибутам, описанным в предыдущем подпункте, процессы могут характеризоваться другими атрибутами, общими для всех процессов. [ISO/IEC 33020: 2015] определяет общие атрибуты процесса, которые характеризуют шесть уровней достижений в рамках системы измерения возможностей процесса. Приложение C включает список атрибутов процесса, которые способствуют достижению более высоких уровней возможностей процесса, как определено в [ISO/IEC 33020: 2015].

5.5.4 Адаптация

Приложение A, которое является нормативным, определяет основные действия, необходимые для выполнения адаптации. Следует отметить, что адаптация может снизить воспринимаемую ценность заявления о соответствии данному документу. Это связано с тем, что другим организациям трудно понять, в какой степени в результате адаптации могут быть удалены желательные положения. Организация, подающая одностороннее заявление о соответствии данному документу, может заявить о полном соответствии меньшему списку процессов, а не об адаптированном соответствии большему списку процессов.

5.6 Группы процессов

5.6.1 Введение

В этом документе действия, которые могут быть выполнены в течение жизненного цикла программной системы, сгруппированы в четыре группы процессов. Каждый из процессов жизненного цикла в этих группах описывается с точки зрения его цели и желаемых результатов с набором связанных действий и задач, которые могут быть выполнены для достижения этих результатов.

Четыре группы процессов и процессы, включенные в каждую группу, изображены на рисунке 4 следующим образом:

a) Процессы согласования;

b) Организационные процессы, способствующие реализации проектов;

c) Процессы технического управления;

d) Технические процессы.

Описанные в данном документе процессы не исключают и не препятствуют использованию дополнительных процессов, которые организации считают полезными. Порядок подразделов, в которых процессы определены в данном документе, не определяет порядок выполнения процессов в течение жизненного цикла системы или любого из его этапов. Описание каждой группы процессов представлено в четырех следующих подпунктах.



**Рисунок 4 – Процессы жизненного цикла программного обеспечения**

5.6.2 Процессы согласования

Организации являются производителями и пользователями программных систем. Одна организация (действующая как покупатель) может поручить другой (действующей как поставщик) товары или услуги. Это достигается с помощью соглашений. Соглашения позволяют покупателям и поставщикам осознавать ценность и поддерживать бизнес-стратегии своих организаций.

Процессы Соглашения – организационные процессы, которые применяются не только в течение жизненного цикла проекта, но и в течение всего срока его действия. Как правило, организации действуют одновременно или последовательно как приобретатели и поставщики программных систем. Процессы Соглашения могут использоваться с меньшей формальностью, если покупатель и поставщик находятся в одной организации. Точно так же они могут использоваться внутри организации для согласования соответствующих обязанностей организации, проекта и технических функций. На рисунке 4 перечислены процессы, входящие в данную группу процессов.

5.6.3 Организационные процессы поддержки проектов

Организационные процессы поддержки проекта связаны с предоставлением ресурсов, позволяющих проекту удовлетворить потребности и ожидания заинтересованных сторон организации. Организационные процессы поддержки проектов обычно связаны на стратегическом уровне с управлением и улучшением бизнеса или предприятия организации, с предоставлением и развертыванием ресурсов и активов, а также с управлением рисками в конкурентных или неопределенных ситуациях. Организационные процессы поддержки проекта применяются вне периода существования проекта, а также в течение всего срока его существования.

Организационные процессы поддержки проектов создают среду, в которой реализуются проекты. Организация устанавливает процессы и модели жизненного цикла, которые будут использоваться проектами; создает, перенаправляет или отменяет проекты; предоставляет необходимые ресурсы, в том числе человеческие и финансовые; и устанавливает и контролирует меры качества для программных систем и других результатов, которые разрабатываются проектами для внутренних и внешних клиентов.

Организационные процессы поддержки проектов создают прочный деловой имидж для многих организаций и предполагают коммерческие и рентабельные мотивы. Тем не менее, организационные процессы поддержки проектов в равной степени актуальны для некоммерческих организаций, поскольку они также подотчетны заинтересованным сторонам, несут ответственность за ресурсы и сталкиваются с рисками в своих начинаниях. Этот документ может применяться как к некоммерческим, так и к коммерческим организациям. На рисунке 4 перечислены процессы, входящие в эту группу процессов.

5.6.4 Процессы технического управления

Процессы технического менеджмента связаны с управлением ресурсами и активами, выделенными руководством организации, и их применением для выполнения соглашений, в которые входит организация или организации. Процессы технического управления связаны с техническими усилиями проектов, в частности, с планированием с точки зрения затрат, сроков и достижений, с проверкой действий, чтобы помочь убедиться, что они соответствуют планам и критериям эффективности, а также с идентификацией и выбором корректирующих действий, которые восполняют недостатки в прогрессе и достижениях. Эти процессы используются для разработки и выполнения технических планов проекта, управления информацией в технической группе, оценки технического прогресса по сравнению с планами для программной системы, продуктов или услуг, контроля технических задач до их завершения и помощи в принятии решений.

Примечание – Технический менеджмент – «применение технических и административных ресурсов для планирования, организации и контроля инженерных функций» (ISO/IEC/IEEE 24765:2010).

Как правило, в одной организации сосуществуют несколько проектов. Процессы технического менеджмента могут быть использованы на корпоративном уровне для удовлетворения внутренних потребностей. На рисунке 4 перечислены процессы, входящие в эту группу процессов.

Примечание – Процессы технического управления применяются во время выполнения каждого технического процесса.

5.6.5 Технические процессы

Технические процессы связаны с техническими действиями на протяжении всего жизненного цикла. Технические процессы преобразуют потребности заинтересованных сторон в продукт или услугу. Применяя этот продукт или эксплуатируя эту услугу, технические процессы обеспечивают устойчивую производительность, когда и где это необходимо для удовлетворения требований заинтересованных сторон и достижения удовлетворенности клиентов. Технические процессы применяются для того, чтобы создать и использовать программную систему, будь она в виде модели или в виде действующего продукта. Технические процессы применяются на любом уровне в иерархии структуры программной системы и на любой стадии жизненного цикла. На рисунке 4 перечислены процессы, входящие в эту группу процессов.

5.7 Применение процесса

Процессы жизненного цикла, определенные в этом документе, могут использоваться любой организацией при приобретении, использовании, создании или поставке программной системы. Они могут применяться на любом уровне иерархии системы и на любом этапе жизненного цикла.

Функции, которые выполняют эти процессы, определены в терминах конкретных целей, результатов и набора видов деятельности и задач, составляющих процесс.

Каждый процесс жизненного цикла на рисунке 4 может быть вызван по мере необходимости в любое время на протяжении всего жизненного цикла. Порядок, в котором процессы представлены в данном документе, не подразумевает какого-либо предписывающего порядка их использования. Однако последовательные отношения вводятся определением модели жизненного цикла. На детальное назначение и время использования процессов на протяжении жизненного цикла влияют многочисленные факторы, включая социальные, торговые, организационные и технические аспекты, каждый из которых может меняться в течение жизни программной системы. Таким образом, индивидуальный жизненный цикл программного обеспечения создается путем выбора и применения процессов, которые обычно обладают параллельными, итеративными, рекурсивными и зависящими от времени характеристиками.

Одновременное использование процессов может существовать как в рамках проекта (например, когда действия по проектированию и подготовительные действия по созданию программной системы выполняются одновременно), так и между проектами (например, когда элементы системы проектируются одновременно в рамках разных проектных обязанностей).

Когда применение одного и того же процесса или набора процессов повторяется в одной и той же системе, такое применение называется итеративным. Итеративное применение процессов важно для постепенного уточнения результатов процесса, например, взаимодействие между последовательными действиями по проверке и интеграции может постепенно укреплять уверенность в соответствии продукта. Итерации не только уместны, но и ожидаемы. Новая информация создается в результате применения процесса или набора процессов. Обычно эта информация принимает форму вопросов в отношении требований, проанализированных рисков или возможностей. Такие вопросы должны быть разрешены до завершения деятельности процесса или набора процессов.

Рекурсивное использование процессов, т.е. повторное применение одного и того же процесса или набора процессов, применяемых к последовательным уровням элементов системы в структуре системы, является ключевым аспектом применения данного документа. Выходы процессов на любом уровне, будь то информация, артефакты или услуги, являются входами для процессов, используемых на уровне ниже (например, при проектировании сверху вниз) или на уровне выше (например, при реализации программной системы). Результаты одного приложения используются в качестве входных данных для следующей более низкой (или более высокой) системы в структуре системы для получения более детального или зрелого набора результатов. Такой подход повышает ценность последующих систем в структуре системы.

Меняющаяся природа воздействий на программную систему (например, изменения операционной среды, новые возможности для реализации элементов системы, изменение структуры и ответственности в организациях) требует постоянного пересмотра выбора и времени использования процессов. Использование процессов в жизненном цикле может быть динамичным, реагируя на многочисленные внешние воздействия на программную систему. Подход, основанный на жизненном цикле, также позволяет учитывать изменения на следующем этапе. Этапы жизненного цикла помогают планировать, выполнять и управлять процессами жизненного цикла в условиях такой сложности жизненных циклов, предоставляя понятные и узнаваемые цели и структуру высокого уровня. Набор процессов в рамках этапа жизненного цикла применяется с общей целью удовлетворения критериев выхода из этого этапа или критериев входа в формальные обзоры хода выполнения работ в рамках этого этапа.

Обсуждение в этом разделе итеративного и рекурсивного использования процессов жизненного цикла программного обеспечения не подразумевает какой-либо конкретной иерархической, вертикальной или горизонтальной структуры для интересующей системы, вспомогательной системы, организации или проекта.

Если это оправдано рисками качества продукции, могут быть также созданы подробные описания экземпляров процессов в контексте конкретной продукции. Инстанцирование процессов включает в себя определение конкретных критериев успеха для экземпляра процесса, полученных из требований к продукции, и определение конкретных видов деятельности и задач, необходимых для достижения критериев успеха, полученных из видов деятельности и задач, определенных в данном документе. Создание подробных описаний экземпляров процессов позволяет лучше управлять рисками качества продукции путем установления связи между процессом и конкретными требованиями к продукции.

Дальнейшее развитие этих концепций можно найти в [ISO/IEC/IEEE 24748 (все части)] по применению процессов жизненного цикла.

5.8 Эталонная модель процесса

Приложение C определяет эталонную модель процесса (PRM) на уровне абстракции выше, чем подробные требования, содержащиеся в пункте 6. PRM применима к организации, которая оценивает свои процессы с целью определения возможностей этих процессов. Цель и результаты представляют собой изложение целей выполнения каждого процесса. Такое изложение целей позволяет оценивать эффективность процессов иными способами, чем простая оценка соответствия.

Примечание – В настоящем документе термин «эталонная модель процесса» используется в том же значении, что и в [ISO/IEC 33001:2015] «модель, включающая определения процессов в области применения, описанные в терминах цели и результатов процесса, вместе с архитектурой, описывающей отношения между процессами».

# 6 Процессы жизненного цикла программного обеспечения

6.1 Процессы согласования

Данный подпункт определяет требования для заключения соглашений с организационными подразделениями, внешними и внутренними по отношению к организации.

Процессы Согласования состоят из перечисленных ниже:

a) Процесс приобретения – используется организациями для приобретения продуктов или услуг;

b) Процесс поставки – используется организациями для поставки продукции или услуг.

Эти процессы определяют действия, необходимые для заключения соглашения между двумя организациями. Если задействован процесс приобретения, он обеспечивает средства для ведения бизнеса с поставщиком. Могут включаться продукты, поставляемые для использования в качестве операционной программной системы, услуги в поддержку операционной деятельности, программные элементы системы или элементы программной системы, предоставляемые поставщиком. Если вызывается процесс поставки, он обеспечивает средства для заключения соглашения, результатом которого является продукт или услуга, предоставляемые приобретателю.

Примечание – Безопасность становится все более важной задачей в области разработки систем и программного обеспечения. Требования и руководство для поставщиков и приобретателей по обеспечению безопасности информации в отношениях с поставщиками приведены в документе [ISO/IEC 27036]. Конкретные аспекты информационной безопасности отношений с поставщиками рассматриваются в [ISO/IEC 27036-3:2013] и [ISO/IEC 27036-4].

6.1.1 Процесс приобретения

6.1.1.1 Цель

Целью процесса приобретения является получение продукта или услуги в соответствии с требованиями приобретателя.

Примечание – В рамках этого процесса соглашение изменяется, когда запрос на изменение, затрагивающий условия соглашения, согласовывается как приобретателем, так и поставщиком.

6.1.1.2 Результаты

В результате успешной реализации процесса приобретения:

a) Подготовлен запрос на поставку.

b) Выбирается один или несколько поставщиков.

c) Между приобретателем и поставщиком заключается соглашение.

d) Принимается продукт или услуга, соответствующие соглашению.

e) Выполнены обязательства приобретателя, определенные в соглашении.

6.1.1.3 Деятельность и задачи

Приобретатель должен выполнять следующие действия и задачи в соответствии с применимыми политиками и процедурами организации в отношении процесса Приобретения.

Примечания

1 Действия и возникающие в результате этого процесса соглашения часто применяются к поставщикам в цепочке поставок, включая субподрядных поставщиков.

2 [IEEE Std 1062-2015] Рекомендуемая практика IEEE для приобретения программного обеспечения, содержит подробные сведения о действиях по альтернативным вариантам приобретения программного обеспечения, включая индивидуально разработанные, готовые к использованию и программное обеспечение как услугу. [IEEE Std 1062-2015] также предоставляет руководство по приобретению программного обеспечения, касающееся технических прав на данные и права интеллектуальной собственности, для обеспечения безопасности или требования к информационной безопасности, а также контрольные списки соответствующих вопросов для рассмотрения в организации при создании процессов приобретения программного обеспечения.

a) Подготовка к приобретению. Эта деятельность состоит из следующих задач:

1) Определение стратегии того, как будет осуществляться приобретение.

Примечания

1 Стратегия описывает или ссылается на модель жизненного цикла, снижение рисков и проблем, график контрольных точек и критерии выбора, если поставщик является внешним по отношению к приобретающей организации. Стратегия также включает ключевые факторы и характеристики приобретения, такие как обязанности и обязательства; конкретные модели, методы или процессы; уровень критичности; формальность; и приоритет соответствующих торговых факторов.

2 Процесс управления решениями и процесс системного анализа часто используются для поддержки компромиссов при выборе стратегии приобретения. Примеры включают: определение решения о производстве или покупке, а также пригодность конкретных COTS или модифицированных решений OTS и оценку поставщика.

3 Если стратегия предусматривает приобретение определенного готового коммерческого программного обеспечения или программного обеспечения с открытым исходным кодом, приобретение может быть ограничено идентификацией поставщика, принятием или согласованием условий в заранее определенной лицензии или соглашении об аренде или обслуживании, определении прав на интеллектуальную собственность и данные в программной системе, а также согласование цены.

4 Факторами, которые следует учитывать в соглашениях между поставщиками, являются права на данные и сторонний доступ к составляющим данным и интеллектуальной собственности. Например, поставщикам одного компонента системы может потребоваться сотрудничество с поставщиками другого компонента и совместное использование исходного кода. Соглашения могут способствовать этому сотрудничеству.

2) Подготовка запроса на поставку продукта или услуги, включающий требования.

Примечания

1 Если поставщик является внешним по отношению к организации, то в запросе указывается деловая практика, которой должен соответствовать поставщик, и критерии выбора поставщика.

2 Определение требований предоставляется одному или нескольким поставщикам. Требования представляют собой требования заинтересованных сторон или требования к системе/программному обеспечению, в зависимости от типа подхода к приобретению и с использованием соответствующего процесса определения требований.

3 Покупатель разрабатывает требования самостоятельно или привлекает для их разработки поставщика. Если приобретатель привлекает поставщика для разработки требований, то приобретатель сохраняет полномочия по утверждению требований, разработанных поставщиком.

b) Объявление о закупке и выбор поставщика. Эта деятельность состоит из следующих задач:

1) передача запроса на поставку продукта или услуги потенциальным поставщикам; и

2) выбор одного или нескольких поставщиков.

Примечание – Для получения конкурентных тендеров предложения на поставку оцениваются и сравниваются с критериями отбора и ранжируются. Обоснование рейтинга каждого предложения объявляется, и поставщики как правило информируются о том, почему они были или не были выбраны.

c) Заключение и поддержание договора. Деятельность состоит из следующих задач:

Примечание – Стоимость, график и производительность проекта отслеживаются в процессе оценки и контроля проекта. Любые выявленные проблемы, требующие изменения соглашения, относятся к этому действию. Любые предложения по изменениям в элементах системы или информации контролируются с помощью операции управления изменениями процесса управления конфигурацией.

1) Разработка соглашения с поставщиком, включающее критерии приемки.

Примечания

1 Формальность этого соглашения варьируется от письменного контракта до устного соглашения. В соответствии с уровнем формальности, соглашение устанавливает требования, этапы разработки и поставки, условия проверки, валидации и приемки, процедуры обработки исключений, процедуры управления изменениями в соглашении и графики платежей, чтобы обе стороны соглашения понимали основу для выполнения соглашения. Другие положения соглашения включают права и ограничения, связанные с техническими данными и интеллектуальной собственностью, подготовку приемочных испытаний и детали испытательной среды; а также степень участия поставщика. В соглашении определяются требования к процессу, которые должны быть предъявлены к участвующим субподрядчикам, например, требования к управлению конфигурацией, отчетность о рисках, отчетность о мерах и анализ измерений.

2 Критерии приемки, такие как приемочные испытания, относятся к тому, как продукт или услуга будут соответствовать своему назначению в условиях эксплуатации. Приемочные испытания могут быть проведены с использованием процесса валидации. Исключения, возникающие в ходе выполнения соглашения или с поставленным продуктом или услугой, разрешаются в соответствии с процедурами, установленными в соглашении.

2) Определение необходимых изменений в соглашении.

Примечание – В запросе на изменение соглашения приобретатель или поставщик подробно излагает свои требования, обоснование и историю вопроса.

3) Оценка влияния изменений на соглашение.

Примечание – Любое изменение исследуется на предмет влияния на планы проекта, график, стоимость, технические возможности и качество. Изменение может быть рассмотрено в рамках существующего соглашения, может потребовать внесения изменений в соглашение или может потребовать заключения нового соглашения.

4) Проведение переговоров о заключении договора с поставщиком.

Примечание – Условия соглашения согласовываются между приобретателем и поставщиком. Переговоры проводятся для первоначального соглашения и по мере необходимости для любых изменений. Измененные соглашения основываются на требуемых изменениях и выявленных последствиях. Детали обсуждаются и изменяются в ходе переговоров, после чего покупатель и поставщик принимают условия соглашения, и оно начинает действовать. Для письменного договора это происходит в момент подписания договора или как указано в соглашении.

5) Обновление договора с поставщиком при необходимости.

Примечания

1 Результат изменения соглашения включается в планы проекта и доводится до сведения всех заинтересованных сторон.

2 В соглашениях могут быть указаны условия, при которых соглашение будет расторгнуто любой из сторон, например, неожиданные изменения в стратегии или доступном финансировании, или отсутствие удовлетворительного прогресса.

a) Мониторинг соглашения. Деятельность состоит из следующих задач:

1) Оценка исполнения соглашения.

Примечания

1 Включает подтверждение того, что все стороны выполняют свои обязанности в соответствии с соглашением. Процесс оценки и контроля проекта используется для оценки прогнозируемой стоимости, графика, производительности и влияния нежелательных результатов на организацию. Информация объединяется с другими оценками выполнения условий соглашения. Если выполнение соглашения не приводит к получению приемлемого продукта или услуги, покупатель или поставщик может расторгнуть соглашение в соответствии с его условиями.

2 Приемочные испытания могут быть проведены с использованием процесса Валидации. Исключения, возникающие во время соглашения или с поставленным продуктом или услугой, разрешаются в соответствии с процедурами, установленными в соглашении.

2) Предоставление данных, необходимых поставщику, и своевременное решение вопросов.

b) Приемка продукта или услуги. Деятельность состоит из следующих задач:

1) Подтверждение соответствия поставленного товара или услуги условиям договора.

Примечание – Если согласованные требования выполнены и критерии приемки соблюдены, поставщик выполнил свое обязательство. Неурегулированные исключения, например, споры по поводу проведения приемочных испытаний или пригодности продукции к использованию по назначению, относятся к положениям соглашения, касающимся споров, арбитража или применимого законодательства и нормативных актов.

2) Предоставление оплаты или другого согласованного вознаграждения.

3) Приемка товара или услуги от поставщика или другой стороны в соответствии с условиями договора.

4) Закрытие договора.

Примечание – Проект закрыт процессом управления проектным портфелем.

6.1.2 Процесс поставки

6.1.2.1 Цель

Цель процесса поставки – предоставить покупателю продукт или услугу, отвечающие согласованным требованиям.

Примечание – В рамках этого процесса соглашение изменяется, когда запрос на изменение, влияющий на условия соглашения, согласовывается как покупателем, так и поставщиком.

6.1.2.2 Результаты

В результате успешного внедрения процесса снабжения:

a) Определяется покупатель продукта или услуги.

b) Составляется ответ на запрос приобретателя.

c) Между приобретателем и поставщиком заключается соглашение.

d) Предоставляется продукт или услуга.

e) Обязательства поставщика, определенные в договоре, выполнены.

Передается ответственность за приобретенный продукт или услугу в соответствии с условиями договора.

6.1.2.3 Деятельность и задачи

Поставщик должен выполнять следующие действия и задачи в соответствии с применимыми политиками и процедурами организации в отношении процесса поставки.

a) Подготовка к поставке. Деятельность состоит из следующих задач:

1) Определение существования и личности приобретателя, у которого есть потребность в продукте или услуге.

Примечание – Данная ситуация часто возникает в процессе анализа бизнеса или миссии. В случае продукта или услуги, разработанной для потребителей, агент, например, отдел маркетинга в организации-поставщика.

2) Определение стратегии поставок.

Примечание – В стратегии описывается или упоминается модель жизненного цикла, снижение рисков и проблем, а также график основных этапов. Она также включает ключевые факторы и характеристики приобретения, такие как обязанности и ответственность; конкретные модели, методы или процессы; уровень критичности; формальность; и приоритет соответствующих торговых факторов.

b) Ответ на запрос о поставке продукции или услуг. Деятельность состоит из следующих задач:

1) Оценка запроса на поставку продукта или услуги) для определения целесообразности и способа реагирования.

2) Подготовка ответа, удовлетворяющего запрос.

c) Заключение и поддержание соглашения. Деятельность состоит из следующих задач:

1) Заключение соглашения с покупателем, которое включает критерии приемки.

Примечание – Формально это соглашение варьируется от письменного до устного. Поставщик подтверждает, что требования, этапы поставки и условия приемки достижимы, что обработка исключений и процедуры управления изменениями соглашения и графики платежей приемлемы, и что они создают основу для выполнения соглашения без ненужных рисков. Вопросы обсуждаются и решаются в ходе переговоров, после чего покупатель и поставщик принимают условия соглашения, и соглашение начинается. Для контракта это происходит при подписании контракта.

2) Определение необходимых изменений в соглашении.

Примечание – Запрашивая изменение соглашения, покупатель или поставщик подробно излагают его спецификации, обоснование и предысторию.

3) Оценка влияния изменений на соглашение

Примечание – Любые изменения исследуются на предмет влияния на планы, график, стоимость, технические возможности или качество проекта. Изменение может быть обработано в рамках существующего соглашения, может потребовать внесения изменений в соглашение или может потребовать нового соглашения.

4) При необходимости обсуждение соглашение с покупателем.

Примечание – Изменения условий соглашения согласовываются между поставщиком и покупателем. Сюда входят изменения, связанные с изменением рыночного контекста. Переговоры происходят для первоначального соглашения и по мере необходимости для любых изменений. Измененные соглашения основаны на необходимых изменениях и выявленных воздействиях.

5) При необходимости обновление договора с эквайером.

Примечание – Результат изменения соглашения включается в планы проекта и доводится до сведения всех затронутых сторон.

d) Оформление договора. Деятельность состоит из следующих задач:

1) Оформление договора согласно установленным планам проекта.

Примечание – Иногда поставщик принимает или соглашается использовать процессы покупателя.

2) Оценка выполнения договора.

Примечание – Включает подтверждение того, что все стороны выполняют свои обязанности в соответствии с соглашением. Процесс оценки и контроля проекта используется для оценки прогнозируемых затрат, графика, производительности и влияния нежелательных результатов на организацию. Действия по управлению изменениями в процессе управления конфигурацией используются для управления изменениями элементов системы. Эта информация сочетается с другими оценками выполнения условий соглашения. Если исполнение соглашения не приводит к получению приемлемого продукта или услуги, покупатель или поставщик может расторгнуть соглашение, как это разрешено в его условиях.

e) Доставка и поддержка продукта или услуги. Деятельность состоит из следующих задач:

1) Доставка товара или услуги в соответствии с критериями соглашения.

Примечание – Как указано в соглашении, критерии приемки, такие как приемочные испытания, относятся к тому, как продукт или услуга будут соответствовать предполагаемому использованию в своей операционной среде. Неурегулированные исключения, например, споры по поводу проведения приемочных испытаний или пригодности продукта для предполагаемого использования, являются предметом положений соглашения, касающихся споров, арбитража или применимого законодательства и нормативных актов.

2) Оказания помощи покупателю в поддержке поставленного продукта или услуги в соответствии с соглашением.

3) Принятие и подтверждение оплаты или иного согласованного вознаграждения.

4) Передача продукта или услуги покупателю или другой стороне, как указано в соглашении.

5) Закрытие договора.

Примечания

1 Проект закрыт процессом управления проектным портфелем.

2 В соглашениях могут быть указаны условия, при которых соглашение будет расторгнуто любой из сторон, например, неожиданные изменения в стратегии или доступном финансировании, или отсутствие удовлетворительного прогресса.

6.2 Организационные процессы поддержки проекта

Организационные процессы поддержки проектов помогают обеспечить способность организации приобретать и поставлять продукты или услуги посредством инициирования, поддержки и контроля проектов. Эти процессы предоставляют ресурсы и инфраструктуру, необходимые для поддержки проектов, и помогают обеспечить выполнение целей организации и установленных соглашений.

Они не предназначены для того, чтобы быть всеобъемлющим набором бизнес-процессов, которые позволяют осуществлять стратегическое управление бизнесом организации.

Организационные процессы поддержки проекта состоят из следующего:

a) Процесс управления моделью жизненного цикла;

b) Процесс управления инфраструктурой;

c) Процесс управления проектным портфелем;

d) Процесс управления человеческими ресурсами;

e) Процесс менеджмента качества;

f) Процесс управления знаниями.

6.2.1 Процесс управления моделью жизненного цикла

6.2.1.1 Цель

Целью процесса «Управление моделями жизненного цикла» является определение, поддержание и обеспечение доступности политик, процессов жизненного цикла, моделей жизненного цикла и процедур для использования организацией в отношении области применения настоящего документа.

Процесс обеспечивает политику жизненного цикла, процессы, модели и процедуры, которые соответствуют целям организации, которые определяются, адаптируются, улучшаются и поддерживаются для поддержки потребностей отдельных проектов в контексте организации, и которые могут применяться с использованием эффективных, проверенных методов и инструментов.

6.2.1.2 Результаты

В результате успешного внедрения процесса управления моделью жизненного цикла:

a) Устанавливаются организационные политики и процедуры для управления и развертывания моделей и процессов жизненного цикла.

b) Определяются ответственность, подотчетность и полномочия в рамках политик, процессов, моделей и процедур жизненного цикла.

c) Оцениваются модели жизненного цикла и процессы, используемые организацией.

d) Реализуются приоритетные улучшения процессов, моделей и процедур.

6.2.1.3 Деятельность и задачи

Организация должна выполнять следующие действия и задачи в соответствии с применимыми политиками и процедурами организации в отношении процесса управления моделью жизненного цикла.

a) Установка процесса. Деятельность состоит из следующих задач:

Примечание – Детализация реализации жизненного цикла в рамках проекта зависит от сложности работы, используемых методов, а также навыков и подготовки персонала, задействованного в выполнении работы. Проект адаптирует политики, процессы, модели и процедуры в соответствии со своими требованиями и потребностями, сохраняя при этом соответствие с правилами и политиками организации. Приложение А содержит информацию о адаптации.

1) Установка политик и процедур для управления процессами и их развертывания, которые согласуются со стратегиями организации.

2) Установка процессов, которые реализуют требования этого документа и согласуются со стратегиями организации.

3) Определение роли, обязанности, подотчетность и полномочия для облегчения реализации процессов и стратегического управления жизненными циклами.

4) Определение бизнес-критерия, который контролирует продвижение по жизненному циклу.

Примечание – Установка критериев принятия решений относительно входа и выхода из каждой стадии жизненного цикла и ключевых этапов. Иногда они выражаются в показателях деловых достижений.

5) Установка стандартных моделей жизненного цикла для организации, которые состоят из этапов, и определение цели и результатов для каждого этапа.

Примечание – Модель жизненного цикла включает одну или несколько моделей этапов, если это необходимо. Он состоит из последовательности этапов, которые могут перекрываться или повторяться в зависимости от масштаба, значимости, сложности, меняющихся потребностей и возможностей интересующей системы. Этапы проиллюстрированы в [ISO/IEC TS 24748-1] с использованием часто встречающегося примера стадий жизненного цикла. Конкретные примеры систем и программного обеспечения приведены в [ISO/IEC TR 24748-2 и ISO/IEC TR 24748]. Процессы и действия жизненного цикла выбираются, приспосабливаются и используются на этапе для достижения цели и результатов этого этапа.

b) Оценка процесса. Деятельность состоит из следующих задач:

Примечание – [ISO/IEC 33002:2015] предоставляет более подробный набор действий и задач по оценке процессов, которые могут быть согласованы с задачами, показанными ниже.

1) Мониторинг выполнения процессов в организации.

Примечание – Включает в себя мониторинг производительности, анализ показателей процесса и анализ тенденций в отношении соблюдения нормативных требований, организационных политик, бизнес-критериев и отзывов от проектов относительно результативности и действенности процессов.

2) Проведение периодических обзоров моделей жизненного цикла, используемых в проектах.

Примечание – Включает в себя подтверждение постоянной пригодности, адекватности и эффективности моделей жизненного цикла, используемых в проектах, и внесение соответствующих улучшений. Сюда входят этапы, процессы и критерии достижения, которые контролируют продвижение по жизненному циклу.

3) Определение возможности улучшения по результатам оценки.

Примечание – Улучшения могут повлиять на этапы, процессы и критерии достижения, которые контролируют продвижение по жизненному циклу.

c) Улучшение процесса. Деятельность состоит из следующих задач:

1) Расстановка приоритетов и планирование возможности улучшения.

2) Реализация возможности улучшения и информирование соответствующих заинтересованных сторон.

Примечание – Улучшение процессов включает улучшения любых процессов в организации. Извлеченные уроки фиксируются и становятся доступны.

6.2.2 Процесс управления инфраструктурой

6.2.2.1 Цель

Целью процесса управления инфраструктурой является предоставление инфраструктуры и услуг проектам для поддержки организации и целей проекта на протяжении всего жизненного цикла.

Процесс определяет, предоставляет и поддерживает средства, инструменты, средства связи и информационные технологии, необходимые для бизнеса организации в соответствии с областью применения этого документа.

6.2.2.2 Результаты

В результате успешного внедрения процесса управления инфраструктурой:

a) Определены требования к инфраструктуре.

b) Элементы инфраструктуры идентифицированы и указаны.

c) Элементы инфраструктуры разработаны или приобретены.

d) Имеется инфраструктура.

6.2.2.3 Действия и задачи

Организация должна выполнять следующие действия и задачи в соответствии с применимыми политиками и процедурами организации в отношении процесса управления инфраструктурой.

a) Создание инфраструктуры. Деятельность состоит из следующих задач:

1) Определение требования к инфраструктуре проекта.

Примечания

1 Примеры элементов инфраструктуры включают средства, инструменты, оборудование, программное обеспечение, услуги и стандарты. В дополнение к общим ресурсам инфраструктуры, общим для организации, для поддержки ее бизнес-процессов, организация также может предоставить проектам уникальные или общие вспомогательные системы для поддержки технических процессов проекта.

2 Потребности в ресурсах инфраструктуры для проекта рассматриваются в контексте с другими проектами и ресурсами внутри организации, а также в рамках политик и стратегических планов организации. Также оцениваются бизнес-ограничения и сроки, которые влияют и контролируют предоставление ресурсов инфраструктуры и услуг для проекта. Планы проектов и будущие потребности бизнеса способствуют пониманию необходимой ресурсной инфраструктуры. Также учитываются физические факторы (например, помещения), потребности в логистике и человеческий фактор (включая аспекты здоровья и безопасности).

3 [ISO/IEC 27036] содержит руководство по обеспечению безопасности инфраструктуры аутсорсинга.

2) Определение, получение и предоставление инфраструктурных ресурсов и услуг, необходимых для реализации и поддержки проектов.

Примечание – Для отслеживания элементов инфраструктуры и поддержки повторного использования инфраструктурных активов часто создается реестр.

b) Поддержка инфраструктуры. Деятельность состоит из следующих задач:

1) Оценка степени соответствия поставляемых инфраструктурных ресурсов потребностям проекта.

2) Выявление и обеспечение улучшений или изменений в ресурсах инфраструктуры по мере изменения требований проекта.

6.2.3 Процесс управления проектным портфелем

6.2.3.1 Цель

Целью процесса управления проектным портфелем является инициирование и поддержание необходимых, достаточных и подходящих проектов для достижения стратегических целей организации.

Процесс обязывает инвестировать достаточное финансирование и ресурсы организации, а также санкционирует полномочия, необходимые для создания выбранных проектов. Он проводит постоянную оценку проектов для подтверждения того, что они оправдывают или могут быть перенаправлены для оправдания дальнейших инвестиций.

Для программных систем управление проектным портфелем также обычно относится к управлению линейкой продуктов (портфелем активов, продуктов и вспомогательных систем или каталогом услуг) для удовлетворения потребностей и целей организации или клиентов и поддержки изменений в технологиях. Управление активами осуществляется через управление проектами.

6.2.3.2 Результаты

Результат успешного внедрения процесса управления проектным портфелем:

a) Квалификация и определение приоритетов возможностей, инвестиций или потребностей в бизнесе.

b) Определены проекты.

c) Выделяются ресурсы и бюджеты для каждого проекта.

d) Определены обязанности, подотчетность и полномочия руководства проекта.

e) Поддерживаются проекты, отвечающие требованиям соглашения и заинтересованных сторон.

f) Проекты, не соответствующие соглашению или не удовлетворяющие требованиям заинтересованных сторон, перенаправляются или прекращаются.

g) Закрываются проекты, по которым завершены соглашения и удовлетворены требования заинтересованных сторон.

6.2.3.3 Деятельность и задачи

Организация должна осуществлять следующие виды деятельности и задачи в соответствии с применимыми политиками и процедурами организации в отношении процесса управления проектным портфелем.

a) Определение и утверждение проектов. Эта деятельность состоит из следующих задач:

1) Определение потенциальных новых или измененных возможностей, или целевых назначений.

Примечание – Бизнес-стратегия организации, концепция операций, анализ изъянов или анализ возможностей рассматриваются на предмет текущих упущений, проблем или возможностей. Новые возможности или потребности предприятия обычно определяются в процессе анализа бизнеса или целевого назначения, далее определяются в процессе определения потребностей и требований заинтересованных сторон и управляются в рамках данного процесса.

2) Определение приоритетов, выбор и создание новых деловых возможностей, начинаний или усилий.

Примечание – Как правило, они соответствуют бизнес-стратегии и планам действий организации. Потенциальным проектам присваивается приоритет, и устанавливаются пороговые значения, чтобы определить, какие проекты будут выполнены. Часто определяются характеристики идентифицированных проектов, включая ценность для заинтересованных сторон, риски и препятствия для успеха, зависимости и взаимосвязи, ограничения, потребности в ресурсах и взаимное соперничество за ресурсы. Затем каждый потенциальный проект оценивается с точки зрения вероятности успеха и соотношения затрат и выгод. Процессы «Управление принятием решений» и «Системный анализ» предоставляют подробную информацию о проведении анализа альтернатив.

3) Определение проектов, ответственности и полномочий.

4) Определение ожидаемых цели, задачи и результатов каждого проекта.

5) Определение и распределение ресурсов для достижения целей и задач проекта.

6) Определение стыков для нескольких проектов и зависимости, которые должны управляться или поддерживаться каждым проектом.

Примечания

1 Включает использование или повторное использование вспомогательных систем, используемых более чем одним проектом, и использование или повторное использование общих элементов системы, включая элементы программного обеспечения, более чем одним проектом.

2 Понимание каждого проекта в контексте архитектуры предприятия помогает обеспечить определение стыков и ограничений.

7) Определение требований к отчетности по проекту и этапы обзора, которые регулируют выполнение каждого проекта.

8) Разрешение каждому проекту приступить к выполнению проектных планов.

Примечание – Анализ процесса планирования проекта для получения дополнительной информации о разработке планов проекта. Планы проектов наиболее полезны, когда они разрабатываются и утверждаются на ранних этапах жизненного цикла проекта.

b) Оценка портфеля проектов. Эта деятельность состоит из следующих задач:

1) Оценка проектов для подтверждения текущей жизнеспособности.

Примечание – Жизнеспособность включает следующие критерии:

i) Проект продвигается к достижению установленных целей и задач.

ii) Проект соответствует директивам проекта.

iii) Проект ведется в соответствии с утвержденными политиками, процессами и процедурами жизненного цикла проекта.

iv) Проект остается жизнеспособным, о чем свидетельствуют, например, сохраняющаяся потребность в услуге, возможность практической реализации продукта и приемлемые инвестиционные преимущества.

2) Действия для продолжения или перенаправления проектов, которые удовлетворительно продвигаются или можно ожидать удовлетворительного продвижения при соответствующем перенаправлении.

c) Прекращение проектов. Эта деятельность состоит из следующих задач:

1) Если позволяют соглашения, принимаются меры по отмене или приостановке проектов, недостатки или риски которых для организации перевешивают выгоды от продолжения инвестиций.

Примечание – Учет уроков, извлеченных из отмененных или неудачных проектов, может быть особенно полезен для совершенствования организации или использования в других проектах.

2) После завершения договора на поставку продукции и услуг, принимаются действия для закрытия проектов.

Примечание – Закрытие осуществляется в соответствии с политикой и процедурами организации, а также соглашением.

6.2.4 Процесс управления человеческими ресурсами

6.2.4.1 Цель

Целью процесса управления человеческими ресурсами является обеспечение организации необходимыми человеческими ресурсами и поддержание их компетенции в соответствии с потребностями бизнеса.

Процесс обеспечивает поставку квалифицированного и опытного персонала, квалифицированного для выполнения процессов жизненного цикла для достижения целей организации, проекта и заинтересованных сторон.

6.2.4.2 Результаты

В результате успешного внедрения процесса управления человеческими ресурсами:

a) Определены навыки, необходимые для реализации проектов.

b) Проекты обеспечены необходимыми человеческими ресурсами.

c) Навыки персонала развиваются, поддерживаются или повышаются.

d) Разрешаются конфликты в требованиях к ресурсам по нескольким проектам.

6.2.4.3 Деятельность и задачи

Организация должна осуществлять следующие виды деятельности и задачи в соответствии с применимыми политиками и процедурами организации в отношении процесса управления человеческими ресурсами:

a) Определение навыков. Эта деятельность состоит из следующих задач:

1) Определение потребности в навыках на основе текущих и ожидаемых проектов.

2) Выявление и регистрация навыков персонала.

b) Развитие навыков. Эта деятельность состоит из следующих задач:

1) Разработка стратегии развития навыков.

Примечание – План включает типы и уровни обучения, категории персонала, графики, потребности в ресурсах для персонала и потребности в обучении.

2) Получение или развитие ресурсов для тренинга, обучения или наставничества.

Примечание – Ресурсы включают учебные материалы, разработанные организацией или внешними сторонами, учебные курсы, доступные у внешних поставщиков, обучение на компьютере. Обеспечить плановое развитие навыков.

3) Ведение учета развития навыков.

c) Приобретение и предоставление навыков. Деятельность состоит из следующих задач:

Примечание – Включает набор и удержание персонала с уровнем опыта и навыков, необходимых для надлежащего укомплектования проектов; оценку и анализ персонала, например, его квалификацию, мотивацию, способность работать в командной среде, а также необходимость переобучения, переназначения или перераспределения кадров.

1) Привлечение квалифицированного персонала при выявлении дефицита навыков.

Примечание – Включает использование сторонних ресурсов.

2) Поддержание и управление резервом квалифицированного персонала, необходимого для выполнения текущих проектов.

3) Назначение задания по проектам с учетом потребностей проекта и развития персонала.

4) Мотивация персонала, например, с помощью механизмов карьерного роста и вознаграждения.

5) Контроль стыков управления несколькими проектами для разрешения конфликтов персонала.

Примечание – Включает в себя конфликты возможностей организационной инфраструктуры и вспомогательных служб и кадровых ресурсов между текущими проектами; или в результате чрезмерной занятости персонала проекта.

6.2.5 Процесс менеджмента качества

6.2.5.1 Цель

Целью процесса управления качеством является обеспечение того, чтобы продукция, услуги и реализация процесса управления качеством соответствовали целям качества организации и проекта и достигали удовлетворенности потребителя.

6.2.5.2 Результаты

В результате успешного внедрения процесса управления качеством:

a) Определены и внедрены политика, цели и процедуры организационного менеджмента качества.

b) Установлены критерии и методы оценки качества.

c) Проектам предоставляются ресурсы и информация для поддержки работы и мониторинга мероприятий по обеспечению качества проекта.

d) Собираются и анализируются результаты оценки обеспечения качества.

e) Политика и процедуры управления качеством совершенствуются на основе результатов проекта и организации.

6.2.5.3 Деятельность и задачи

Организация должна осуществлять следующие виды деятельности и задачи в соответствии с применимыми политиками и процедурами организации в отношении процесса менеджмента качества.

Примечание – Информация и требования к созданию системы менеджмента качества приведены в [ISO 9001:2015].

a) Планирование управления качеством. Деятельность состоит из следующих задач:

1) Установить политику, цели и процедуры управления качеством.

Примечания

1 [ISO 9004:2009] включает руководящие принципы по улучшению производительности.

2 Политика, цели и процедуры основаны на бизнес-стратегии, направленной на удовлетворение потребностей клиентов и управление рисками.

2) Определить ответственность и полномочия для внедрения менеджмента качества.

Примечание – Ресурсы для управления качеством могут выделяться из отдельных организаций для независимости от управления проектом.

3) Определение критериев и методов оценки качества.

4) Предоставление ресурсов и информации для управления качеством.

b) Оценка менеджмента качества. Деятельность состоит из следующих задач:

1) Сбор и анализ результатов оценки обеспечения качества в соответствии с определенными критериями.

2) Оценка удовлетворённости клиента.

Примечание – [ISO 10004: 2012] включает рекомендации по мониторингу и измерению удовлетворенности клиентов. Качество программного обеспечения также подтверждается удовлетворенностью клиентов.

3) Проводить периодические обзоры деятельности по обеспечению качества проекта на предмет соответствия политике, целям и процедурам управления качеством.

Примечание – Должны быть установлены критерии и методы оценки качества. Оценка качества касается соблюдения процедур проекта и соответствия продукции характеристикам качества.

4) Отслеживание состояние улучшения качества процессов, продуктов и услуг.

c) Выполнение корректирующих и предупреждающих действий. Деятельность состоит из следующих задач:

1) Планирование корректирующих действий, если не достигнуты цели менеджмента качества.

2) Планирование предупреждающих действий, когда существует достаточный риск того, что цели менеджмента качества не будут достигнуты.

3) Отслеживание корректирующих и предупреждающих действий до их завершения и информирование заинтересованной стороны.

Примечания

1 Реализация корректирующих и предупреждающих действий осуществляется в рамках других соответствующих процессов, таких как «Управление моделью жизненного цикла» или «Оценка и контроль проекта».

2 [ISO 9001:2015, 0.3.3] и Приложение A.4 описывают превентивные действия для устранения потенциальных несоответствий, как часть риск-ориентированного мышления.

6.2.6 Процесс управления знаниями

6.2.6.1 Цель

Целью процесса управления знаниями является создание возможностей и активов, которые позволяют организации использовать возможности для повторного применения существующих знаний.

Сюда входят знания, навыки и активы знаний, включая элементы системы.

Примечание – Повторное применение существующих знаний известно как повторное использование знаний и включает повторное использование знаний об элементах программного обеспечения или полученных из них.

6.2.6.2 Результаты

В результате успешного внедрения процесса управления знаниями:

a) Определена систематическая классификация для применения активов знаний.

b) Развиваются или приобретаются организационные знания, навыки и активы знаний.

c) Имеются организационные знания, навыки и активы знаний.

d) Данные об использовании управления знаниями собираются и анализируются.

6.2.6.3 Деятельность и задачи

Организация должна выполнять следующие действия и задачи в соответствии с применимыми политиками и процедурами организации в отношении процесса управления знаниями:

a) Планирование управления знаниями. Эта деятельность состоит из следующих задач:

1) Определение стратегии управления знаниями

Примечания

1 Cтратегия управления знаниями обычно включает:

i) Определение областей и их потенциала для повторного применения знаний.

ii) Планы по получению и поддержанию знаний, навыков и активов знаний в течение срока их полезного использования.

iii) Характеристика типов знаний, навыков и активов знаний, которые необходимо собирать и поддерживать.

iv) Критерии принятия, квалификации и отказа от знаний, навыков и активов знаний.

v) Процедуры контроля изменений в знаниях, навыках и активах знаний.

vi) Планы, механизмы и процедуры защиты, контроля и доступа к секретным или конфиденциальным данным и информации.

vii) Механизмы хранения и поиска.

2 Управление знаниями включает в себя знания, которыми обмениваются как внутри организации, так и знания, которыми обмениваются за пределами организации с заинтересованными сторонами, покупателями и деловыми партнерами в соответствии с соглашениями об интеллектуальной собственности и неразглашении.

2) Определение знаний, навыков и активов знаний, которыми нужно управлять.

3) Определение проектов, которые могут выиграть от применения знаний, навыков и активов знаний.

b) Распространение знаний и навыков в рамках всей организации. Эта деятельность состоит из следующих задач:

1) Создание и поддержка классификации для сбора и обмена знаниями и навыками в организации.

Примечание – Классификация может включать экспертные, общие и предметные знания и навыки, а также уроки, извлеченные из других задач.

2) Овладение или приобретение знаний и навыков.

3) Распространение знаний и навыков в организации.

c) Распространение активов знаний в рамках всей организации. Эта деятельность состоит из следующих задач:

1) Создание систематической квалификации (таксономии) для организации активов знаний.

Примечания

1 Таксономия включает следующее:

i) Определение границ предметных областей и их отношения к другим предметным областям.

ii) Модели предметной области, отражающие основные общие и различные функции, возможности, концепции и функции.

iii) Архитектуру семейства систем в предметной области, включая их общие и различные функции.

2 [ISO/IEC 26550] применяется для получения дополнительной информации о моделях продуктовой линейки. [ISO/IEC/IEEE 42010:2011] применяется для определения требований к архитектурным каркасам, точкам зрения, видам моделей, представлениям и моделям.

2) Разработка или приобретение активов знаний.

Примечание – Активы знаний включают элементы системы или их представления (например, библиотеки многократно используемого кода, эталонные архитектуры), архитектуру или элементы дизайна (например, архитектуру или шаблоны дизайна), процессы, критерии или другую техническую информацию (например, учебные материалы), относящуюся к знаниям области, и извлеченные уроки.

3) Обмен знаниями в рамках организации.

Примечание – Автоматизированные возможности поиска улучшают доступ к ресурсам знаний.

d) Управление знаниями, навыками и активами знаний. Деятельность состоит из следующих задач:

1) Поддержка знаний, навыков и активов знаний.

2) Мониторинг и регистрация повторного использования знаний, навыков и активов знаний.

3) Периодическое проведение переоценки актуальности технологий и потребностей рынка в активах знаний.

Примечание – Оценка преимущества для бизнеса, которые организация получила благодаря использованию практики управления знаниями.

6.3 Процессы технического менеджмента

Процессы технического менеджмента используются для создания и развития планов, выполнения планов, оценки фактических достижений и прогресса в сравнении с планами, а также для контроля выполнения вплоть до реализации. Отдельные процессы технического менеджмента могут быть задействованы в любой момент жизненного цикла и на любом уровне иерархии проектов, как того требуют планы или непредвиденные события. Процессы технического менеджмента применяются с той степенью строгости и формальности, которая зависит от риска и сложности проекта.

Область применения процесса технического менеджмента – техническое управление проектом или его продуктами, включая программный продукт или интересующую систему.

Примечание – Набор процессов технического управления применяется для эффективного выполнения технических процессов, связанных с программной системой. Они не включают в себя систему управления или исчерпывающий набор процессов для управления проектами, так как это не входит в объем настоящего документа.

Процессы технического менеджмента включают в себя следующее:

a) Процесс планирования проекта;

b) Процесс оценки и контроля проекта;

c) Процесс управления принятием решений;

d) Процесс управления рисками;

e) Процесс управления конфигурацией;

f) Процесс управления информацией;

g) Процесс измерения;

h) Процесс обеспечения качества.

Процессы планирования проекта и оценки и контроля проекта являются ключевыми для всех методов управления. Эти процессы устанавливают общий подход к управлению проектом или процессом. Другие процессы этой группы обеспечивают конкретный сфокусированный набор задач для достижения специализированной цели управления. Все они проявляются в управлении любым предприятием, начиная от полной организации и заканчивая отдельным процессом жизненного цикла и его задачами. В данном документе в качестве контекста для описания процессов выбран проект. Эти же процессы могут быть применены и при оказании услуг.

6.3.1 Процесс планирования проекта

6.3.1.1 Цель

Целью процесса планирования проекта является создание и координация эффективных и работоспособных планов.

Процесс определяет объем деятельности по управлению проектом и технической деятельности, определяет результаты процесса, задачи и результаты, устанавливает графики выполнения задач, включая критерии достижения, и необходимые ресурсы для выполнения задач. Это непрерывный процесс, который продолжается в течение всего проекта, с регулярным пересмотром планов.

Примечание – Стратегии, определенные в каждом из других процессов, обеспечивают входные данные и интегрируются в процесс планирования проекта. Процесс оценки и контроля проекта используется для оценки того, являются ли планы интегрированными, согласованными и выполнимыми.

6.3.1.2 Результаты

В результате успешного внедрения процесса планирования проекта:

a) Определены цели и планы.

b) Определены роли, обязанности, подотчетность и полномочия.

c) Ресурсы и услуги, необходимые для достижения целей, официально запрашиваются и предоставляются.

d) Активизированы планы по выполнению проекта.

6.3.1.3 Деятельность и задачи

В рамках проекта должны быть выполнены следующие мероприятия и задачи в соответствии с применимыми политиками и процедурами организации в отношении процесса планирования проекта.

a) Определение проекта. Деятельность состоит из следующих задач:

1) Определение цели и ограничений проекта.

Примечания

1 Цели и ограничения включают производительность и другие аспекты качества, стоимость, время и удовлетворенность клиентов и пользователей. Каждая цель определяется с уровнем детализации, позволяющим выбрать, адаптировать и реализовать соответствующие процессы и мероприятия.

2 [ISO/IEC 15026, ISO/IEC 27001 и ISO/IEC 27036] предоставляют дополнительное руководство по целям и ограничениям, связанным с обеспечением и безопасностью.

2) Определение объема проекта, установленного в соглашении.

Примечание – Сюда входят соответствующие действия, необходимые для удовлетворения критериев бизнес-решений и успешного завершения проекта. Проект может отвечать за один или несколько этапов полного жизненного цикла программной системы. Планирование проекта включает определение соответствующих действий по ведению планов проекта, выполнению оценок и контролю проекта.

3) Определение и поддержание модели жизненного цикла, состоящей из этапов, используя определенные модели жизненного цикла организации.

Примечание – [ISO/IEC TS 24748-1] содержит подробную информацию о стадиях жизненного цикла и определении соответствующей модели жизненного цикла. Он определяет общий набор этапов жизненного цикла образцовой системы, включая концепцию, разработку, производство, использование, поддержку и списание. Он также определяет общий примерный набор этапов жизненного цикла программного обеспечения, включая определение потребностей, исследование и определение концепции, демонстрацию и оценку, проектирование/разработку, производство/изготовление, развертывание/продажи, эксплуатацию, обслуживание и поддержку и списание.

4) Создание структуры разбивки работ (WBS) на основе поставляемых продуктов или развивающейся архитектуры программной системы.

Примечания

1 Каждый элемент архитектуры программной системы, а также соответствующие процессы и виды деятельности описываются с уровнем детализации, соответствующим выявленным рискам. Связанные задачи в структуре разбивки работ группируются для выполнения. Задачи проекта определяют рабочие элементы, которые разрабатываются или производятся. Стандарт управления практикой для структур разбивки работ Института управления проектами (PMI) содержит дополнительные подробности о WBS.

2 В проектах с гибкими или итерационными методами элемент WBS может соответствовать основным характеристикам, с точки зрения пользователя, которые должны быть созданы в течение итераций.

5) Определение и поддержка процессов, которые будут применяться в проекте.

Примечания

1 Процессы основаны на определенных процессах организации (например, процесс управления моделью жизненного цикла). Приложение A содержит информацию по адаптации, которая может быть использована для удовлетворения специфических потребностей проекта. Определение процессов включает критерии входа и выхода, входы, ограничения последовательности процессов (отношения предшественник/преемник), требования к параллельности процессов (какие процессы и задачи должны выполняться одновременно с другими задачами или видами деятельности в области процессов), атрибуты «Показатели эффективности/Показатели результативности», а также параметры объема и стоимости (для критически важной оценки стоимости).

2 Определение стыков с другими проектами или организационными подразделениями решается в процессе управления портфелем проектов.

b) Планирование управления проектами и техническими средствами. Эта деятельность состоит из следующих задач:

1) Определение и ведение графика проекта на основе управленческих и технических целей и сметы работ.

Примечание – Это включает определение продолжительности, взаимосвязи, зависимостей и последовательности работ, этапов достижения, задействованных ресурсов, а также обзоров и резервов графика для управления рисками, необходимых для своевременного завершения проекта.

2) Определение критериев достижения для критериев принятия решений по этапам жизненного цикла, сроки выполнения и основные зависимости от внешних входов или выходов.

Примечание – Временные интервалы между внутренними проверками определяются в соответствии с политикой организации по таким вопросам, как критичность бизнеса и системы, график и технические риски.

3) Определение расходов и планирование бюджета.

Примечание – Бюджетные затраты основаны на графике, оценках размера и сложности программного обеспечения, оценках трудозатрат, затратах на инфраструктуру, статьях закупок, оценках приобретенных услуг и вспомогательных систем, а также бюджетных резервах для управления рисками.

4) Определение роли, обязанностей, подотчетности и полномочий.

Примечание – Это включает определение организации проекта, набор персонала и развитие навыков персонала. К полномочным органам относятся, при необходимости, юридически ответственные роли и лица, например, авторизация проектирования, авторизация безопасности, а также лица, ответственные за применимые сертификаты или аккредитации.

5) Определение необходимой инфраструктуры и необходимых услуг.

Примечание – Это включает определение необходимых мощностей, их наличия и распределения по задачам проекта. Инфраструктура включает помещения, услуги, инструменты, средства связи и информационные технологии. Также указаны требования к вспомогательным системам и сервисам для каждой стадии жизненного цикла.

6) Планирование приобретения материалов и вспомогательных систем и услуг, поставляемых из-за пределов проекта.

Примечания

1 Это включает, при необходимости, планы запроса предложений, выбора поставщика, принятия, администрирования контракта и закрытия контракта. Процессы соглашения используются для запланированных приобретений.

2 [ISO/IEC 27036] содержит руководство для приобретения инфраструктуры и услуг.

7) Создание и передача плана проекта, техническое управление и исполнение, включая обзоры.

Примечания

1 Техническое планирование для программной системы часто отражается в плане управления системной инженерией (SEMP), плане управления программной инженерией или плане разработки программного обеспечения (SDP). [ISO/IEC/IEEE 24748-5] более подробно описывает планирование технического управления программной инженерией и включает снабжённый комментариями план SDP. Планирование проекта часто отражается в Плане управления проектом. [ISO/IEC/IEEE 16326] содержит более подробную информацию о планировании проекта.

2 Стратегические мероприятия и задачи из каждого из других процессов обеспечивают входные данные и интегрируются в процесс планирования проекта. Процесс оценки и контроля проекта используется для обеспечения интеграции, согласованности и выполнимости планов.

c) Активация проекта. Деятельность состоит из следующих задач:

1) Получение разрешения на начало проекта.

Примечание – Одобрение начала работ (разрешение на продолжение работ) предоставляется в рамках процесса управления портфелем проектов.

2) Подача заявок и получение обязательств на необходимые ресурсы для выполнения проекта.

Примечание – Это включает в себя доступ к вспомогательным системам или услугам.

3) Реализация планов проекта.

6.3.2 Процесс оценки и контроля проекта

6.3.2.1 Цель

Целью процесса оценки и контроля проекта является оценка соответствия и выполнимости планов; определение статуса проекта, технических и технологических показателей; и руководство выполнением, чтобы помочь обеспечить выполнение в соответствии с планами и графиками, в рамках прогнозируемых бюджетов, для удовлетворения технических целей.

Процесс периодически и на крупных мероприятиях оценивает прогресс и достижения в соответствии с требованиями, планами и общими бизнес-целями. Информация предоставляется для действий руководства при обнаружении значительных отклонений. Этот процесс также включает перенаправление деятельности и задач проекта, если это необходимо, для исправления выявленных отклонений и отклонений от других технических процессов управления или технических процессов. Перенаправление может включать перепланировку в зависимости от обстоятельств.

6.3.2.2 Результаты

В результате успешного внедрения процесса оценки и контроля проекта:

a) Имеются показатели эффективности или результаты оценки.

b) Оценивается адекватность ролей, обязанностей, подотчетности и полномочий.

c) Оценивается достаточность ресурсов.

d) Проводятся обзоры технического прогресса.

e) Изучаются и анализируются отклонения в выполнении проекта от планов.

f) Заинтересованные стороны информируются о состоянии проекта.

g) Определяются и направляются корректирующие действия, если результаты проекта не соответствуют целевым показателям.

h) При необходимости инициируется перепланирование проекта.

i) Санкционируются действия по продвижению (или не продвижению) проекта от одной запланированной вехи или события к следующей.

j) Цели проекта достигнуты.

6.3.2.3 Деятельность и задачи

В рамках проекта должны быть выполнены следующие мероприятия и задачи в соответствии с применимыми политиками и процедурами организации в отношении процесса оценки и контроля проекта.

a) План оценки и контроля проекта. Деятельность состоит из следующих задач:

1) Определение стратегии оценки и контроля проекта.

Примечание – Стратегия определяет ожидаемые мероприятия по оценке и контролю проекта, включая запланированные методы оценки и сроки, а также необходимые управленческие и технические обзоры.

b) Оценка проекта. Эта деятельность состоит из следующих задач:

1) Оценка соответствия целей и планов проекта контексту проекта.

2) Оценка планов руководства и технических планов в сравнении с целями для определения адекватности и осуществимости.

3) Оценка состояния проекта и технического состояния в сравнении с соответствующими планами для определения фактических и прогнозируемых отклонений в стоимости, графике и производительности.

4) Оценка адекватности ролей, обязанностей, подотчетности и полномочий.

Примечание – Это включает оценку адекватности компетенций персонала для выполнения проектных ролей и выполнения задач проекта. По возможности используются объективные показатели, например, эффективность использования ресурсов, достижения проекта.

5) Оценка адекватности и доступности ресурсов.

Примечание – Ресурсы включают инфраструктуру, персонал, финансирование, время или другие соответствующие элементы. Эта задача включает оценку повторного использования существующих процессов и ресурсов инфраструктуры, а также подтверждение того, что внутриорганизационные обязательства выполнены.

6) Оценка прогресса с использованием измеренных достижений и прохождений этапов.

Примечание – Это включает сбор и оценку данных о стоимости труда, материалов, услуг и технических характеристик, а также других технических данных о целях, таких как доступность. Эти данные сравниваются с показателями достижения целей. Это включает в себя проведение оценки эффективности для определения адекватности развивающейся программной системы требованиям. Это также включает в себя готовность вспомогательных систем предоставлять свои услуги, когда это необходимо.

7) Проведение необходимых управленческих и технических обзоров, аудитов и инспекций.

Примечание – Они бывают формальными или неформальными и проводятся для определения готовности к переходу на следующий этап жизненного цикла или вехи проекта, для обеспечения выполнения проектных и технических задач или для получения обратной связи от заинтересованных сторон.

8) Мониторинг критических процессов и новых технологий.

Примечание – Это включает в себя определение и оценку зрелости технологии и целесообразности ее внедрения. Зрелость технологии - это готовность технологии к оперативному использованию, и часто измеряется по шкале от низкой (существует только в виде концепции) до высокой (доказано оперативное использование).

9) Анализ результатов измерений и рекомендации.

Примечание – Результаты измерений анализируются для выявления отклонений, вариаций или нежелательных тенденций от запланированных значений, которые включают потенциальные проблемы, и для выработки соответствующих рекомендаций по исправлению или профилактическим действиям. Это включает, при необходимости, статистический анализ мер, указывающих на тенденции, например, плотность отказов, указывающая на качество выходных данных, распределение измеренных параметров, указывающее на повторяемость процесса.

10) Запись и предоставление статуса и результатов выполнения заданий по оценке.

Примечание – Как правило, они указываются в соглашении, политиках и процедурах.

11) Мониторинг выполнения процессов в рамках проекта.

Примечание – Это включает анализ показателей процесса и обзор тенденций в отношении целей проекта. Любые выявленные действия по улучшению могут быть выполнены в рамках процесса обеспечения качества, процесса управления качеством или процесса управления моделью жизненного цикла.

c) Контроль проекта. Деятельность состоит из следующих задач:

1) Инициация необходимых действий для решения выявленных проблем.

Примечания

1 Задача возникает, когда проект или технические достижения не соответствуют запланированным показателям. Сюда входят профилактические, корректирующие действия и действия по решению проблем. Действия обычно требуют перепланирования или переназначения персонала, инструментов и инфраструктурных активов, когда обнаружено несоответствие или недоступность, или когда проект или технические достижения превышают цели или план. Они часто влияют на стоимость, график, технический объем или толкование. Действия иногда требуют внесения изменений в реализацию и выполнение.

2 Действия регистрируются и пересматриваются для подтверждения их адекватности и своевременности.

2) Инициация необходимого перепланирования проекта.

Примечания

1 Перепланирование проекта инициируется, когда цели проекта или ограничения изменились, или когда предположения планирования оказались несостоятельными.

2 Любое изменение, требующее внесения изменений в соглашение между приобретателем и поставщиком, задействует процессы «Приобретение» и «Поставка».

3) Инициация действия по внесению изменений в случае изменения стоимости, времени или качества в соответствии с контрактом из-за воздействия запроса покупателя или поставщика.

Примечание – Это включает в себя рассмотрение измененных условий поставки или инициирование выбора нового поставщика, что задействует процессы закупок и поставок.

4) Разрешение проекту перейти к следующему этапу или мероприятию, если это оправдано.

Примечание – Процесс оценки и контроля проекта используется для достижения соглашения о завершении основных этапов.

6.3.3 Процесс управления решениями

6.3.3.1 Цель

Целью процесса управления принятием решений является обеспечение структурированной, аналитической основы для объективного определения, характеристики и оценки набора альтернатив для принятия решения в любой точке жизненного цикла и выбора наиболее выгодного образа действий.

Процесс используется для решения технических или проектных вопросов и ответов на запросы о принятии решений, возникающих в течение жизненного цикла программного обеспечения, с целью определения альтернативы (альтернатив), обеспечивающей предпочтительные результаты для данной ситуации. Методы, наиболее часто используемые для управления решениями, – торговое исследование и инженерный анализ. Каждая из альтернатив оценивается по критериям принятия решения (например, влияние на стоимость, влияние на график, программные ограничения, нормативные последствия, технические характеристики, критические характеристики качества и риск). Результаты этих сравнений ранжируются с помощью подходящей модели выбора и затем используются для принятия оптимального решения. Ключевые данные исследования (например, допущения и обоснование решения) обычно сохраняются для информирования лиц, принимающих решения, и поддержки принятия решений в будущем.

Примечание – Когда необходимо провести детальную оценку параметра по одному из критериев, для проведения оценки может быть использован процесс системного анализа.

6.3.3.2 Результаты

В результате успешного внедрения процесса управления принятием решений:

a) Определяются решения, требующие альтернативного анализа.

b) Определяются и оцениваются альтернативные варианты действий.

c) Выбирается предпочтительный курс действий.

d) Определяется решение, обоснование решения и допущения.

6.3.3.3 Деятельность и задачи

В рамках проекта должны быть выполнены следующие мероприятия и задачи в соответствии с применимыми политиками и процедурами организации в отношении процесса управления принятием решений.

a) Подготовка к принятию решений. Деятельность состоит из следующих задач:

1) Определение стратегии управления решениями.

Примечание – Стратегия управления принятием решений включает в себя определение ролей, обязанностей, подотчетности и полномочий. Стратегия учитывает необходимость получения входной информации и своевременного принятия решения. Она включает определение категорий решений и схему расстановки приоритетов. Решения часто возникают в результате оценки эффективности, технического компромисса, проблемы, требующей решения, действия, необходимого в ответ на риск, превышающий допустимый порог, или новой возможности, или одобрения перехода проекта на следующий этап жизненного цикла. Руководство организации или проекта определяет степень строгости и формальности анализа решений.

2) Определение обстоятельства и необходимости принятия решения.

Примечание – Проблемы или возможности и альтернативные варианты действий, которые позволят решить их, регистрируются, классифицируются и сообщаются.

3) Привлечение соответствующих заинтересованных сторон к принятию решений с целью использования опыта и знаний.

Примечание – Эффективная практика заключается в определении экспертных знаний, необходимых для анализа и принятия решения.

b) Анализ информации о принятом решении. Деятельность состоит из следующих задач:

1) Выбор и объявление стратегии управления для каждого решения.

Примечание – Определяется степень строгости, необходимая для решения этих проблем или возможностей, а также данные и системный анализ, необходимые для оценки альтернатив. Определите временные рамки для принятия решения.

2) Определение желаемых результатов и измеримых критериев отбора.

Примечание – Определяется желаемое значение для количественных критериев и пороговое(ые) значение(я), за пределами которого атрибут будет неудовлетворительным, а также весовые коэффициенты для критериев.

3) Определение торгового пространства и альтернативы.

Примечание – Если существует большое количество альтернатив, они подвергаются качественному отбору, чтобы сократить их число до приемлемого для дальнейшего детального системного анализа. Отбор часто основан на качественных оценках таких факторов, как риск, стоимость, график и нормативное воздействие.

4) Оценка каждой альтернативы по критериям.

Примечание – Процесс системного анализа используется, при необходимости, для количественной оценки конкретных критериев для каждой торговой альтернативы, подлежащей оценке. Это включает новые параметры конструкции, различные характеристики архитектуры и диапазон значений для критических характеристик качества. В процессе системного анализа оценивается диапазон изменения параметров, чтобы получить анализ чувствительности для каждого из оцениваемых торговых альтернатив. Эти результаты используются для определения осуществимости различных торговых альтернатив.

c) Приняие решений и управление ими. Деятельность состоит из следующих задач:

1) Определение предпочтительной альтернативы для каждого решения.

Примечание – Альтернативы оцениваются количественно с использованием критериев выбора. Выбранная альтернатива, как правило, обеспечивает оптимизацию или улучшение определенного решения.

2) Запись решения, обоснование решения и предположения.

3) Регистрация, отслеживание, оценка и сообщение о принятых решениях.

Примечания

1 Включает в себя записи о проблемах и возможностях и их решении, как это предусмотрено соглашениями или организационными процедурами, и в форме, позволяющей проводить аудит и учиться на опыте.

2 Позволяет организации подтвердить, что проблемы были эффективно решены, что неблагоприятные тенденции были обращены вспять, и что были использованы преимущества возможностей.

6.3.4 Процесс управления рисками

6.3.4.1 Цель

Целью процесса управления рисками является постоянное выявление, анализ, лечение и мониторинг рисков.

Процесс управления рисками – непрерывный процесс систематического устранения рисков на протяжении всего жизненного цикла системного продукта или услуги. Он может быть применен к рискам, связанным с приобретением, разработкой, обслуживанием или эксплуатацией системы.

Примечание – Риск определяется в [ISO Guide 73:2009] как «Влияние неопределенности на цели». К нему прилагается Примечание 1: «Эффект – отклонение от ожидаемого – положительное и/или отрицательное». Положительный риск обычно называют возможностью, и он может быть рассмотрен в рамках процесса управления рисками.

6.3.4.2 Результаты

В результате успешного внедрения процесса управления рисками:

a) Риски определены.

b) Риски проанализированы.

c) Варианты работы с рисками выявлены, расставлены по приоритетам и определены.

d) Соответствующая обработка применяется.

e) Риски оценены для определения изменений в статусе и прогресса в обработке.

6.3.4.3 Деятельность и задачи

Проект должен реализовать следующие мероприятия и задачи в соответствии с применимыми политиками и процедурами организации в отношении процесса управления рисками.

Примечание – [ISO/IEC/IEEE 16085] предоставляет более подробный набор мероприятий и задач по управлению рисками. Данный процесс управления рисками согласован с [ISO 31000:2009 и ISO Guide 73:2009]. В стандарте [ISO 9001:2015] планирование рисков и возможностей описано в п. 6.1.

a) План управления рисками. Деятельность состоит из следующих задач:

1) Определение стратегии управления рисками.

Примечание – Включает процесс управления рисками поставщиков цепочки поставок и описывает, как риски от поставщиков будут подниматься на следующий уровень (уровни) для включения в процесс управления рисками проекта.

2) Определение и запись контекста процесса управления рисками.

Примечания

1 Включает описание перспектив заинтересованных сторон, категорий риска, а также описание (возможно, со ссылкой) технических и управленческих целей, допущений и ограничений. Категории риска включают соответствующие технические области программной системы и облегчают идентификацию рисков на протяжении всего жизненного цикла продукта. Как указано в стандарте [ISO 31000], цель этапа – составить полный список рисков на основе тех событий, которые могут создать, усилить, предотвратить, ухудшить, ускорить или задержать достижение целей.

2 Возможности, которые являются одним из видов риска, обеспечивают потенциальные выгоды для программной системы или проекта. Каждая из реализуемых возможностей имеет сопутствующие риски, которые отвлекают от ожидаемой выгоды. Сюда входят риски, связанные с отказом от реализации возможности, а также риск не достижения эффекта от реализации возможности.

b) Управление профилем риска. Деятельность состоит из следующих задач:

1) Определение и фиксирование порогового значения риска и условия, при которых уровень риска может быть приемлем.

2) Создание и поддержание профиля риска.

Примечание – Профиль риска фиксирует: контекст управления рисками; запись состояния каждого риска, включая вероятность его возникновения, последствия и пороговые значения риска; приоритет каждого риска на основе критериев риска, предоставленных заинтересованными сторонами; и запросы на действия по риску вместе со статусом их обработки. Профиль риска обновляется, когда происходят изменения в состоянии отдельного риска. Приоритет в профиле риска используется для определения применения ресурсов для обработки.

3) Периодическое предоставление соответствующего профиля риска заинтересованным сторонам в соответствии с их потребностями.

c) Анализ рисков. Деятельность состоит из следующих задач:

1) Идентификация рисков в категориях, описанных в контексте управления рисками.

Примечание – Риски обычно выявляются в ходе различных анализов, таких как анализ безопасности, надежности, безопасности и производительности, оценка технологии, архитектуры и готовности, а также торговые исследования. Риски часто выявляются на ранних этапах жизненного цикла и сохраняются в процессе использования, поддержки и списания программной системы. Кроме того, риски часто выявляются в ходе анализа измерений развивающейся программной системы.

2) Оценка вероятности возникновения и последствия каждого выявленного риска.

Примечание – Последствия риска обычно включают в себя технические, календарные, стоимостные или качественные последствия.

3) Оценка каждого риска в соответствии с пороговыми значениями риска.

4) Для каждого риска, который не соответствует порогу риска, определение и фиксирование рекомендуемых стратегий и мер обработки.

Примечание – Стратегии обработки риска включают, но не ограничиваются устранением риска, снижением вероятности его возникновения или тяжести последствий, или принятием риска. Обработка также включает принятие или увеличение риска для того, чтобы реализовать возможность. Меры предоставляют информацию об эффективности альтернативных вариантов обработки.

d) Обработка рисков. Деятельность состоит из следующих задач:

1) Определение рекомендуемых альтернатив для обработки рисков.

2) Реализация альтернативных вариантов обработки риска, для которых заинтересованные стороны определяют необходимость принятия действий, чтобы сделать риск приемлемым.

3) Постоянный контроль риска, который не соответствует пороговому значению и принятый заинтересованными сторонами, определение его как высокоприоритетного и определение необходимости в дальнейших действиях по обработке риска , если изменился его приоритет.

4) Координирование действий по управлению риском после выбора способа по его устранения.

Примечание – Можно применить процесс оценки и контроля проекта.

e) Мониторинг рисков. Деятельность состоит из следующих задач:

1) Постоянное отслеживание рисков и контекста управления рисками на предмет изменений, оценивание рисков при их изменении.

2) Внедрение и контролирование мер по оценке эффективности устранения рисков.

3) Постоянный мониторинг появления новых рисков и источников на протяжении всего жизненного цикла.

6.3.5 Процесс управления конфигурацией

6.3.5.1 Цель

Целью управления конфигурацией является управление и контроль элементов и конфигураций системы на протяжении всего жизненного цикла. Управление конфигурацией (CM) также управляет согласованностью между продуктом и связанным с ним определением конфигурации.

Управление конфигурацией программного обеспечения (SCM) применяется как к программной системе, так и к ее интерфейсам. Целью управления интерфейсами является согласование с заинтересованными сторонами по интерфейсам обмена данными посредством коммуникаций между программными системами и сервисами. В приложении E (п. E.5) приведен пример представления процесса управления интерфейсами.

Конфигурации программного обеспечения изменяются посредством контролируемого выпуска новой версии. Цель выпуска – санкционировать и обеспечение доступности программной возможности, функции или системы для определенной цели, с ограничениями или без ограничений для подмножества пользователей.

6.3.5.2 Результаты

В результате успешного внедрения процесса управления конфигурацией:

a) Определяются и управляются элементы, требующие управления конфигурацией.

b) Устанавливаются базовые параметры конфигурации.

c) Изменения элементов, находящихся под управлением конфигурации, контролируются.

d) Доступна информация о состоянии конфигурации.

e) Проводятся необходимые аудиты конфигурации.

f) Релизы и поставки системы контролируются и утверждаются.

6.3.5.3 Деятельность и задачи

Проект должен реализовать следующие мероприятия и задачи в соответствии с применимыми политиками и процедурами организации в отношении процесса управления конфигурацией.

Примечание – [ISO/IEC/IEEE 19770] содержит процедуры и требования к системе управления ИТ-активами.

a) Планирование управления конфигурацией. Эта деятельность состоит из следующих задач:

1) Определение стратегии управления конфигурацией, включая подходы для следующего:

i) Управление CM, включая роли, обязанности, подотчетность и полномочия, а также использование плат контроля конфигурации (контроля изменений);

ii) Рассмотрение уровня риска и воздействия при утверждении базовых параметров конфигурации и запросов на регулярные и аварийные изменения.

Примечание – Переодически планируемые изменения для применения исправлений программного обеспечения с использованием утвержденных процедур или регистрации и возврата тестируемых программных элементов, находящихся в стадии разработки, как правило выполняются автоматически или проверяются и утверждаются ежедневно в установленном порядке. Для сравнения: значительные изменения в конструкции системы программного обеспечения, оказывающие серьезное влияние на стоимость и график проекта, могут включать в себя обширный анализ, консультации с поставщиками, обзоры заинтересованных сторон и утверждения на высших уровнях организации.

iii) Координация CM по всей совокупности организаций приобретателя, поставщика и цепочки поставок в течение всего срока службы программной системы или в рамках соглашения или проекта, в зависимости от обстоятельств.

iv) Контроль доступа, изменений и распоряжения элементами конфигурации.

v) Необходимые исходные данные, которые должны быть установлены, включая критерии или события для начала контроля конфигурации и поддержания исходных данных изменяющихся конфигураций.

vi) Контроль лицензий на программное обеспечение, прав на данные и других активов интеллектуальной собственности.

vii) Частота, приоритеты и содержание версий и релизов программного обеспечения.

viii) Стратегия аудита и обязанности по проверке непрерывной целостности и безопасности информации об определении конфигурации

ix) Управление изменениями, включая подготовку заинтересованных сторон и особенно пользователей к изменениям в операционных программных системах и услугах.

Примечания

1 Для сложных программных систем проводятся исследования компромиссов, например, для выбора соответствующего автоматизированного инструмента для поддержки потребностей и объема SCM, определенных в стратегии.

2 Дополнительное руководство по управлению конфигурацией приведено в [ISO 10007, IEEE Std 828 и SAE ANSI/EIA-649-B].

3 [SWEBOK, Руководство к своду знаний по программной инженерии] содержит подробное описание SCM. Здесь рассматривается SCM в контексте системы, планирование проекта и процесса SCM, план и схема SCM, выбор инструментов, контроль субподрядчиков, наблюдение и другие аудиты, элементы и отношения конфигурации программного обеспечения, библиотеки программного обеспечения и деятельность процесса управления конфигурацией.

4 Стратегия SCM обычно документируется в плане, например, в плане управления конфигурацией, или иногда в SEMP, SDP или плане управления проектом (PMP). Планирование стратегии управления конфигурацией координируется в процессе планирования проекта. При определении точек для установления базовых показателей и проведения аудитов планирование УУК согласуется с жизненным циклом программного обеспечения. Частота повторяющихся мероприятий SCM соответствует итерациям технических процессов и стадий. Планирование SCM обычно включает решение о том, когда пересматривать планирование управления конфигурацией, какие условия требуют обновления плана CM, и кто уполномочен изменять планы CM и элементы, находящиеся под контролем конфигурации.

2) Определение процедуры хранения, архивирования и извлечения для элементов конфигурации, артефактов КМ и записей.

Примечание – Места и условия хранения элементов конфигурации программной системы, таких как исходный код и исполняемое программное обеспечение, устанавливаются в соответствии с установленными уровнями целостности, безопасности и сохранности.

b) Выполнение идентификации конфигурации. Деятельность состоит из следующих задач:

1) Выбор элементов программной системы, которые должны быть однозначно идентифицированы как элементы конфигурации, подлежащие конфигурационному контролю.

***Пример – Элементы конфигурации, подлежащие контролю конфигурации в программных системах, обычно включают спецификации требований к системе/программному обеспечению, спецификации интерфейса; элементы продукта и системы (например, программные объекты, аппаратные средства и услуги) в процессе разработки, базовые конфигурации или версии программного обеспечения, установленные для перехода между этапами и после выпуска в оперативное использование; основные («золотые») копии кода источника или исполняемого программного обеспечения для различных платформ или версий; конфигурации для конкретного объекта в оперативном использовании; и информационные элементы, такие как соглашения, модели архитектуры, описания услуг и операционных процедур; и элементы в вспомогательных системах.***

Примечания

1 Уникальная идентификация может быть применена к компонентам программного обеспечения, версиям или индивидуально лицензированным копиям. Идентификаторы соответствуют стандартам и отраслевым соглашениям, так что объекты, находящиеся под контролем конфигурации, однозначно прослеживаются до их поставщика и до их спецификаций или эквивалентных записанных описаний. Информационные элементы часто идентифицируются и управляются отдельно от других элементов конфигурации.

2 Идентификаторы конфигурации программного обеспечения облегчают отслеживание, когда несколько разработчиков или программистов сопровождения работают над одной и той же функцией программного обеспечения, так что различные ветви кода могут быть успешно собраны и протестированы.

3 [ISO/IEC 19770 (все части)] предусматривает систему управления ИТ-активами для отслеживания лицензий на программное обеспечение.

2) Определение атрибутов элементов конфигурации.

Примечания

1 Атрибуты относятся к статусу элемента, физическим или логическим характеристикам, полезным для управления или обслуживания программной системы. Соответствующие атрибуты могут различаться для элементов конфигурации аппаратного и программного обеспечения.

2 Атрибуты и идентификаторы конфигурации могут отражать декомпозицию программной системы, так что элементы конфигурации отслеживаются на уровне, на котором необходимо контролировать изменения.

***Пример – Программное обеспечение от внешних поставщиков может отслеживаться по лицензионному соглашению и соглашению о техническом обслуживании, что может включать отслеживание местоположения, количества или размера систем, где оно используется, или количества одновременно работающих пользователей. Версии программного обеспечения можно отследить по требованиям заинтересованных сторон, которые они реализуют.***

3) Определение базовых линий на протяжении всего жизненного цикла.

Примечания

1 Базовые параметры фиксируют изменяющиеся состояния конфигурации элементов программной системы в определенное время или при определенных обстоятельствах. Содержание базовых линий разрабатывается в рамках технических процессов, но формализуется в определенный момент времени в рамках процесса управления конфигурацией. Базовые параметры формируют основу для последующих изменений. Выбранные базовые параметры обычно формализуются между приобретателем и поставщиком, в зависимости от практики отрасли и договорного участия приобретателя в процессе управления конфигурацией. На системном уровне обычно существует три основных типа базовых линий: функциональная базовая линия, выделенная базовая линия и базовая линия продукта. Они различаются в зависимости от области или местной стратегии.

2 Разработка программных систем часто подразумевает создание нескольких базовых линий развития для удовлетворения меняющихся потребностей в конфигурации программного обеспечения в ключевые моменты жизненного цикла, например, для обеспечения одновременного контроля версий программного обеспечения во время разработки программного обеспечения, прототипов, интеграции и тестовых выпусков. Это может включать распределение обязанностей по управлению конфигурацией и ограничение доступа к архивам, например, библиотекам разработки или тестирования программного обеспечения и библиотеке поддержки основной конфигурации.

4) Получение согласия покупателя и поставщика на установление базового уровня.

Примечание – Для достижения соглашения используется процесс оценки и контроля проекта. Когда программное обеспечение разрабатывается для коммерческого или внутреннего использования, приобретатель или спонсор проекта может быть уполномочен утвердить базовую линию.

c) Управление изменениями конфигурации. Деятельность состоит из следующих задач:

Примечание – Управление изменениями конфигурации устанавливает процедуры и методы для управления изменениями базовой линии после ее создания. Иногда это называют управлением конфигурацией. Термин «управление изменениями» также используется для управления изменениями в организационных процедурах и рабочих процессах.

1) Определение и регистрация запросов на изменение и запросов на отклонение от нормы.

Примечание – Запрос на отклонение часто называют отклонением, отказом или уступкой.

2) Координация, оценка и отклонение запросов на изменение и запросов на отклонение от нормы.

Примечания

1 Оценка обычно включает анализ обоснования и необходимости в сравнении с воздействием на программное обеспечение и взаимодействующие системы, учитывая риски и возможности, качество, пользователей, график и стоимость. Принимается решение о реализации или отклонении запроса на изменение.

2 Запросы на изменение и запросы на отклонение часто находятся под формальным контролем Совета по контролю конфигурации (CCB).

3) Отслеживание и управление утвержденными изменениями базовой линии, запросами на изменение и запросами на отклонение.

Примечания

1 Задача включает в себя определение приоритетов, отслеживание, планирование и закрытие изменений. Затем изменения вносятся в рамках технических процессов. Эти изменения проверяются или подтверждаются в рамках процессов верификации и валидации, чтобы убедиться, что утвержденные изменения были применены правильно.

2 Изменения и обоснования, как правило, регистрируются при утверждении и при завершении работы.

d) Выполнение контроля выпуска. Деятельность состоит из следующих задач:

1) Идентификация и регистрация запросов на выпуск, с определением элементов программной системы в выпуске.

Примечание – Модель жизненного цикла помогает определить частоту итеративных или инкрементных выпусков программного обеспечения. Процесс интеграции используется для выбора и настройки пакета выпуска, версии программного обеспечения, обновления или исправления. Эти изменения проверяются или подтверждаются с помощью процессов Верификации или Валидации. Изменения вносятся с помощью технических процессов, в частности процесса перехода.

***Пример – Релизы для испытания программного обеспечения, для квалификации программного обеспечения или системы или других формальных испытаний, или для пробного (бета) или эксплуатационного использования.***

2) Утверждение релизов и поставок программных систем.

Примечания

1 Релизы как правило включают в себя определение приоритетов, отслеживание, планирование и закрытие изменений. Утверждение релиза для оперативного использования может включать принятие проверенных и подтвержденных изменений. Критерии утверждения релиза как правило включают планы возврата или планы действий в чрезвычайных ситуациях в случае неудачного релиза.

2 Для программных систем автоматизированные средства контроля версий могут помочь обеспечить доступ к только правильным версиям исходного кода, их обновление, тестирование и документирование утвержденных изменений соответствующим персоналом, а также их выпуск.

3) Отслеживание и управление распределением релизов программных систем по заданным средам или поставкам программного обеспечения.

Примечание – Основные копии или копии инкрементных изменений выпущенных версий программного обеспечения могут поддерживаться в течение всего срока службы системы или проекта в контролируемой среде. Поставщики программного обеспечения часто отслеживают доставленные копии лицензионного программного обеспечения до приобретателя, чтобы обеспечить согласованное сопровождение программного обеспечения. Релиз программной системы хранится и распространяется в соответствии с соглашением и политикой.

e) Выполнение учета состояния конфигурации. Деятельность состоит из следующих задач:

1) Разработка и поддержка информации о состоянии CM для элементов программных систем, базовых линий и релизов.

Примечания

1 Учет состояния конфигурации предоставляет данные о состоянии контролируемых продуктов, необходимые для принятия решений относительно элементов системы на протяжении всего жизненного цикла продукта. Например, статус программного обеспечения может включать прошлое, текущее и планируемое продвижение по этапам жизненного цикла для функций программного обеспечения и завершение мероприятий по верификации и валидации для элементов программного обеспечения. Информация о состоянии конфигурации позволяет проследить прямую и обратную связь с другими состояниями конфигурации. Записи о состоянии конфигурации ведутся в течение жизненного цикла программного обеспечения или проекта, а затем архивируются в соответствии с соглашениями, соответствующим законодательством или практикой организации.

2 Управление записью, извлечением и консолидацией текущего состояния конфигурации и состояния предыдущих конфигураций для подтверждения корректности, своевременности, целостности и безопасности информации. Аудиты проводятся для проверки соответствия базовой линии представлению архитектуры, документам по управлению интерфейсом, лицензионным соглашениям на программное обеспечение и другим требованиям соглашения.

2) Сбор, хранение и отчетность по данным управления конфигурацией.

f) Выполнение оценки конфигурации. Деятельность состоит из следующих задач:

1) Определение необходимости проведения аудитов CM и составление графика проведения мероприятий.

2) Проверка соответствия конфигурации продукта требованиям к конфигурации путем сравнения требований, ограничений и отступлений (отклонений) с результатами формальных мероприятий по проверке, которые могут включать методы выборки.

3) Контроль внесения утвержденных изменений в конфигурацию.

4) Оценка соответствия программной системе функциональным и эксплуатационным возможностям, определенным для базовой линии.

Примечание – Иногда это называется аудитом функциональной конфигурации (FCA), который гарантирует, что конфигурация продукта соответствует установленным требованиям.

5) Оценка соответствия элементов операционной программной системы утвержденной информации о конфигурации.

Примечание – Иногда это называется аудитом физической конфигурации (PCA). Для элементов программного обеспечения критерии PCA могут включать в себя оценку следующего: установлены ли указанные элементы конфигурации на означенных системах в соответствии с лицензией или они установлены по соглашению на программное обеспечение.

6) Фиксация результатов аудита CM и пункты действий по устранению недостатков.

6.3.6 Процесс управления информацией

6.3.6.1 Цель

Целью процесса управления информацией является генерирование, получение, подтверждение, преобразование, сохранение, извлечение, распространение и утилизация информации для назначенных заинтересованных сторон.

Управление информацией планирует, выполняет и контролирует предоставление информации заинтересованным сторонам, которая является однозначной, полной, проверяемой, последовательной, модифицируемой, отслеживаемой и презентабельной. Информация включает в себя техническую, проектную, организационную, договорную и пользовательскую информацию. Информация часто берется из записей данных организации, системы, процесса или проекта.

Примечание – Управляемая информация имеет такие характеристики качества: однозначность, полнота, проверяемость, последовательность, изменяемость, прослеживаемость и презентабельность.

6.3.6.2 Результаты

В результате успешного внедрения процесса управления информацией:

a) Определяется информация, которой необходимо управлять.

b) Определяются представления информации.

c) Информация получена, разработана, преобразована, сохранена, подтверждена, представлена и утилизирована.

d) Определяется статус информации.

e) Информация доступна заинтересованным сторонам.

6.3.6.3 Деятельность и задачи

Проект должен реализовать следующие мероприятия и задачи в соответствии с применимыми политиками и процедурами организации в отношении процесса управления информацией.

Примечание – [ISO/IEC/IEEE 15289] обобщает требования к содержанию информационных элементов процессов жизненного цикла (документации) и предоставляет руководство по их разработке.

a) Подготовка к управлению информацией. Эта деятельность состоит из следующих задач:

1) Определение стратегии управления информацией.

Примечание – Информация по одной и той же теме может быть разработана разными способами на разных этапах жизненного цикла и для разных аудиторий.

2) Определение элементов информации, которые будут управляться.

Примечание – Сюда входит информация, которая будет управляться в течение жизненного цикла программного обеспечения и, возможно, поддерживаться в течение определенного периода после него. Это делается в соответствии с политикой организации, соглашениями или законодательством.

3) Назначение органов и ответственных по управлению информацией.

Примечание – Должное внимание уделяется законодательству в области информации и данных, безопасности и конфиденциальности, например, права собственности, ограничения по соглашению, права доступа к данным и права собственности на данные, интеллектуальная собственность и патенты. Там, где действуют ограничения или запреты, информация идентифицируется соответствующим образом. Сотрудники, владеющие такой информацией, информируются о своих обязательствах и ответственности.

4) Определение содержания, формата и структуры информационных элементов.

Примечание – Информация возникает и заканчивается в различных формах (например, аудиовизуальной, текстовой, графической, числовой) и носителях (например, электронных, печатных, магнитных, оптических). Учитываются организационные ограничения, например, инфраструктура, межорганизационные связи и распределенная работа над проектом. Соответствующие стандарты и соглашения по информационным элементам используются в соответствии с политикой, соглашениями и законодательными ограничениями.

5) Определение действий по обслуживанию информации.

Примечание – Обслуживание информации включает в себя проверку состояния хранимой информации на предмет целостности, достоверности и доступности. Оно также включает любые потребности в репликации или преобразовании на альтернативный носитель, по мере необходимости, либо для сохранения инфраструктуры при изменении технологии, чтобы архивные носители можно было читать, либо для переноса архивных носителей на более новые технологии.

b) Осуществление управления информацией. Деятельность состоит из следующих задач:

1) Получение, разработка или преобразование идентифицированных элементов информации.

Примечание – Включает сбор данных, информации или информационных элементов из соответствующих источников (например, в результате любого процесса жизненного цикла), а также написание, иллюстрирование или преобразование их в информацию, пригодную для использования заинтересованными сторонами. Она включает в себя проверку, подтверждение и редактирование информации в соответствии с информационными стандартами.

2) Ведение информационных объектов и учет их хранения, а также регистрация состояния информации.

Примечания

1 Информационные элементы поддерживаются в соответствии с требованиями к их целостности, безопасности и конфиденциальности. Поддерживается статус информационных элементов (например, описание версии, дата выпуска или срок действия, запись о распространении, классификация безопасности). Легко читаемая информация хранится и сохраняется таким образом, чтобы ее можно было легко найти.

2 Исходные данные и инструменты, используемые для преобразования информации, а также полученная документация помещаются под контроль конфигурации в соответствии с процессом управления конфигурацией. [ISO/IEC/IEEE 26531] содержит требования к системам управления контентом, полезным для информации и документации жизненного цикла.

3) Публикация, распространение или предоставление доступа к информации и информационным материалам заинтересованным сторонам.

Примечание – Информация предоставляется заинтересованным сторонам в соответствующей форме, как того требуют согласованные графики или определенные обстоятельства. Информационные элементы включают документацию, используемую для сертификации, аккредитации, лицензирования или оценки рейтингов, в соответствии с требованиями.

4) Архивирование обозначенной информации.

Примечание – Архивирование осуществляется в соответствии с целями аудита, сохранения знаний и закрытия проекта. Носители, местоположение и защита информации выбираются в соответствии с установленными сроками хранения и поиска, а также политикой организации, соглашениями и законодательством. Принимаются меры для сохранения необходимых информационных элементов после закрытия проекта.

5) Утилизация нежелательной, недействительной или непроверенной информации.

Примечание – Это делается в соответствии с политикой организации и требованиями безопасности и конфиденциальности.

6.3.7 Процесс оценки

6.3.7.1 Цель

Целью процесса оценки является сбор, анализ и предоставление объективных данных и информации для поддержки эффективного управления и демонстрации качества продукции, услуг и процессов.

Примечание – Оценка обладает такими качественными характеристиками: проверяемость, значимость, возможность принятия мер, своевременность и экономическая эффективность.

6.3.7.2 Результаты

В результате успешного внедрения процесса Оценки:

a) Определяются информационные потребности.

b) Определяется или разрабатывается соответствующий набор оценок, основанный на информационных потребностях.

c) Необходимые данные собираются, проверяются и хранятся.

d) Данные анализируются, а результаты интерпретируются.

e) Информационные элементы предоставляют объективную информацию, которая поддерживает принятие решений.

6.3.7.3 Деятельность и задачи

Проект должен реализовать следующие мероприятия и задачи в соответствии с применимыми политиками и процедурами организации в отношении процесса Оценки.

Примечание

1 [ISO/IEC 15939] предоставляет более подробный набор оценочных мероприятий и задач, которые соответствуют мероприятиям и задачам в данном документе.

2 [ISO 9001:2015] устанавливает требования к системе менеджмента качества для оценки и мониторинга.

a) Подготовка к оценке. Деятельность состоит из следующих задач:

1) Определение стратегии оценки.

2) Описание характеристики организации, имеющие отношение к оценке, такие как деловые и технические цели.

3) Определение и расстановка приоритетов информационных потребностей.

Примечание – Информационные потребности основаны на бизнес-целях организации, целях проекта, выявленных рисках и других моментах, связанных с проектными решениями. Оценки могут относиться к проектам, процессам, продуктам или решениям.

4) Выбор и уточнение оценок, удовлетворяющих информационные потребности.

Примечание – Определены оценки, которые поддаются проверке и являются экономически эффективными.

5) Определение процедуры сбора, анализа, доступа и отчетности.

6) Определение критериев для оценки информационных элементов и процесса Оценки.

7) Определение и планирование необходимых вспомогательных систем или услуг для использования.

b) Выполнение оценки. Деятельность состоит из следующих задач:

1) Интеграция ручных или автоматизированных процедур генерации данных, сбора, анализа и отчетности в соответствующие процессы.

Примечание – Эта задача может включать влияние изменений на другие процессы жизненного цикла для выполнения процедурной интеграции.

2) Сбор, хранение и проверка данных.

3) Анализ данных и разработка информационных материалов.

4) Регистрация результатов и информирование пользователей оценок.

Примечание – Результаты анализа оценок сообщаются соответствующим заинтересованным сторонам своевременно и в удобной для использования форме для поддержки принятия решений и помощи в корректирующих действиях, управлении рисками и улучшениях. Результаты сообщаются участникам процесса принятия решений, участникам технической и управленческой экспертизы, а также владельцам процессов совершенствования продукции и процессов.

6.3.8 Процесс обеспечения качества

6.3.8.1 Цель

Цель процесса обеспечения качества – помочь обеспечить эффективное применение процесса управления качеством организации к проекту.

Обеспечение качества фокусируется на обеспечении уверенности в том, что требования к качеству будут выполнены. Опережающий анализ процессов и результатов жизненного цикла проекта проводится для обеспечения требуемого качества производимого продукта и соблюдения политики и процедур организации и проекта.

6.3.8.2 Результаты

В результате успешного внедрения процесса обеспечения качества:

a) Определены и внедрены процедуры обеспечения качества проекта.

b) Определены критерии и методы оценки обеспечения качества.

c) Оценка продукции, услуг и процессов проекта проводится в соответствии с политикой, процедурами и требованиями менеджмента качества.

d) Результаты оценок предоставляются соответствующим заинтересованным сторонам.

e) Устранение инцидентов.

f) Приоритетные проблемы обрабатываются

Примечание – Результаты a) – d) соответствуют результатам деятельности и задач процесса управления качеством.

6.3.8.3 Деятельность и задачи

В рамках проекта должны быть выполнены следующие мероприятия и задачи в соответствии с применимыми политиками и процедурами организации в отношении процесса обеспечения качества.

Примечание – [IEEE Std 730-2014] содержит дополнительную информацию.

a) Подготовка к обеспечению качества. Эта деятельность состоит из следующих задач:

1) Определение стратегии обеспечения качества. Стратегия соответствует политике и целям управления качеством организации и включает в себя:

i) Приоритеты для применения ресурсов обеспечения качества к процессам и задачам, которые оказывают наиболее значительное влияние на качество поставляемой продукции и услуг;

ii) Определенные роли, обязанности, подотчетность и полномочия;

iii) Критерии и методы оценки процессов, продукции и услуг, включая критерии приемки продукции или услуг;

iv) Мероприятия, соответствующие каждому поставщику (включая субподрядчиков);

v) необходимые мероприятия по проверке, валидации, мониторингу, измерению, обзору, инспекции, аудиту и тестированию, специфичные для продукции или услуг; и

vi) Решение проблем и деятельность по совершенствованию процессов и продукции.

Примечание – В проектах программного обеспечения деятельность и задачи, оказывающие значительное влияние на качество продукции, включают получение согласия по новым и измененным требованиям, проведение экспертных оценок и модульного тестирования, анализ отчетов о проблемах и отзывов пользователей; подтверждение завершения корректирующих действий, назначенных на контрольных этапах проекта, и анализ первопричин дефектов.

2) Установка независимости обеспечения качества от других процессов жизненного цикла.

Примечание – Ресурсы для обеспечения качества часто выделяются из отдельных организаций для независимости от управления проектом.

b) Проведение оценки продукции или услуг. Деятельность состоит из следующих задач:

1) Оценка продуктов и услуг на соответствие установленным критериям, контрактам, стандартам и нормам.

Примечание – Эта задача включает проверку того, отражены ли критерии приемки продукции или услуг в деятельности по верификации и валидации. Производные требования к качеству системы/программного обеспечения обычно связаны с характеристиками качества в процессе определения требований. [ISO/IEC 25010, ISO/IEC 25030] предоставляют дополнительную информацию о характеристиках качества системы/программного обеспечения.

2) Контроль выполнения верификации и валидации результатов процессов жизненного цикла для определения соответствия заданным требованиям.

c) Проведение оценки процессов. Эта деятельность состоит из следующих задач:

1) Оценка процессов жизненного цикла проекта на предмет их соответствия.

2) Оценка инструментов и сред, которые поддерживают или автоматизируют процесс обеспечения соответствия.

3) Оценка процессов поставщика на соответствие требованиям к процессам.

Примечание – Рассматриваются такие вопросы, как совместная среда разработки программного обеспечения, меры по процессу, которые поставщики обязаны предоставить, или процесс оценки рисков, который поставщики обязаны использовать. Это включает в себя надзорные обзоры внедрения процесса в цепочке поставок.

d) Управление записями и отчетами по обеспечению качества. Деятельность состоит из следующих задач:

1) Создание записей и отчетов, связанных с деятельностью по обеспечению качества.

Примечание – Создание записи и отчетов в соответствии с организационными, нормативными и проектными требованиями, с использованием процесса управления информацией.

2) Ведение, хранение и распространение записей и отчетов.

3) Выявление инцидентов и проблем, связанных с оценками продуктов, услуг и процессов.

Примечание – Включает в себя фиксацию извлеченных уроков. Определены обязанности по разрешению.

e) Обработка инцидентов и проблем. Деятельность состоит из следующих задач:

Примечания

1 В терминологии менеджмента качества проблемы часто описываются как «несоответствия», когда, если их не устранить, они могут привести к тому, что проект не сможет удовлетворить его требованиям.

2 Дополнительная информация и примеры категорий проблем и классификации приоритетов приведена в [ISO/IEC TS 24748-1:2016, Приложение C].

1) Регистрация, анализ и классификация инцидентов.

2) Определение выбранных инцидентов, чтобы связать их с известными ошибками или проблемами.

3) Запись, анализ и классификация проблем.

Примечание – Результаты анализа включают потенциальные варианты обработки.

4) Выявление коренных причин и обработка проблем там, где это возможно.

5) Определение приоритетов в решении проблем (разрешение проблем) и отслеживание корректирующих действий.

Примечание – Внедрение осуществляется в технических процессах после инициации процесса оценки и контроля проекта. Организационные процедуры эскалации проблем могут помочь сфокусировать ресурсы на запаздывающих решениях проблем.

6) Анализ тенденций в области инцидентов и проблем.

7) Выявление улучшения в процессах и продуктах, которые могут предотвратить будущие инциденты и проблемы.

Примечание – Процесс управления рисками используется для обработки рисков и возможностей. Процесс управления моделью жизненного цикла используется для улучшения процессов организации.

8) Информирование заинтересованных сторон о статусе инцидентов и проблем.

9) Отслеживание инцидентов и проблем до их устранения.

6.4 Технические процессы

Технические процессы используются для определения требований к программной системе, преобразования требований в эффективный продукт, обеспечения последовательного воспроизведения продукта, когда это необходимо, использования продукта для предоставления необходимых услуг, поддержания предоставления этих услуг и утилизации продукта, когда он выводится из эксплуатации.

Технические процессы определяют деятельность, которая позволяет организационным и проектным функциям оптимизировать выгоды и снизить риски, возникающие в результате технических решений и действий. Эта деятельность позволяет программным системам и услугам обладать своевременностью и доступностью, экономической эффективностью, функциональностью, надежностью, ремонтопригодностью, возможностью производства, удобством использования и другими качествами, требуемыми приобретающими и поставляющими организациями. Они также позволяют продуктам и услугам соответствовать ожиданиям или законодательно установленным требованиям общества, включая здоровье, безопасность, надежность и экологические факторы.

Технические процессы включают в себя следующее:

a) процесс анализа бизнеса или цели работы;

b) процесс определения потребностей и требований заинтересованных сторон;

c) процесс определения требований к системе/программному обеспечению;

d) процесс определения архитектуры;

e) процесс определения проектного решения;

f) процесс системного анализа;

g) процесс внедрения;

h) процесс интеграции;

i) процесс верификации;

j) процесс перехода;

k) процесс валидации;

l) процесс эксплуатации;

m) процесс технического обслуживания;

n) процесс утилизации.

Примечания

1 Для программных систем эти процессы могут быть рекурсивно применены на более широком или более детальном уровнях для определения и реализации программной системы.

2 Для программных систем процессы часто выполняются параллельно, итерационно сменяя друг друга для создания решения, которое имеет удовлетворительные компромиссы в отношении требований, критических показателей производительности и критических характеристик качества. На любом уровне требования и модели приводятся в соответствие посредством итераций применимых технических процессов. Когда требования и модели не могут быть реализованы напрямую, технические процессы применяются рекурсивно на более детальном уровне или через различные представления системы.

3 Концепция этапов жизненного цикла и применение этих процессов на любом этапе подробно описаны в [ISO/IEC TS 24748-1]. В нем есть полный набор примеров этапов и результатов этапов для применения технических процессов в рамках жизненного цикла программного обеспечения.

4 Управление интерфейсами – набор видов деятельности, которые пересекаются с процессами программной инженерии. Сквозные виды деятельности технологических процессов и процессов технического управления применяются и отслеживаются как специфическое представление процессов и программной системы. Пример представления процесса управления интерфейсами приведен в Приложении E (E.5).

5 [ISO/IEC 27002, ISO/IEC 27034] предоставляют руководство по применению вопросов безопасности в технических процессах для программных систем. Например, Приложение E (E.6) для представления процесса обеспечения безопасности программного обеспечения.

6.4.1 Процесс анализа бизнеса или цели работы

6.4.1.1 Цель

Целью процесса анализа бизнеса или цели работы является определение проблемы или возможности бизнеса или цели работы, характеристика пространства решений и определение потенциального класса (классов) решений, которые могут решить проблему или воспользоваться возможностью.

Примечания

1 Анализ бизнеса и цели работы связан с организацией, охватывающей заинтересованные стороны, которые имеют отношение к деятельности жизненного цикла программного обеспечения. Процесс взаимодействует со стратегией организации, которая обычно находится вне области применения [ISO/IEC/IEEE 12207]. Результаты стратегического анализа организации включают концепцию деятельности организации, стратегические цели и планы, новые элементы рынка или миссии, а также выявленные проблемы и возможности. Стратегия организации устанавливает контекст, в котором проводится анализ бизнеса или миссии. Концепция деятельности организации относится к предполагаемом руководством способу функционирования организации. Она описывает предположения организации и то, как она намерена использовать, приобретать или поставлять разрабатываемую систему, существующие системы и возможные будущие системы в поддержку общей операции или серии операций предприятия. В случае, когда организация является системой, представляющей интерес, стратегия организации является частью определения системы.

2 Процесс применяется на протяжении всего срока службы программного системного решения и может быть пересмотрен в случае изменений в окружающей среде, потребностях или других факторах.

3 В некоторых областях анализ бизнеса или цели работы относится к концепции выявления и анализа возможностей, которые необходимы или желательны для организации. Этот процесс фокусируется на необходимых возможностях и взаимодействует с процессом управления проектным портфелем для определения торгового пространства, которое может использовать эти возможности. Выявленные проблемы или возможности часто переводятся в целевые возможности. В зависимости от того, что применимо в данной области, проблема или пространство возможностей включает в себя целевые возможности.

6.4.1.2 Результаты

В результате успешного внедрения процесса анализа бизнеса или цели работы:

a) Определяется проблема или пространство возможностей.

b) Характеризуется пространство решений.

c) Определяются предварительные операционные концепции и другие концепции на этапах жизненного цикла.

d) Определяются и анализируются альтернативные классы решений-кандидатов.

e) Выбирается предпочтительный(ые) класс(ы) альтернативных вариантов решений.

f) Доступны любые вспомогательные системы или услуги, необходимые для анализа бизнеса или цели работы.

g) Устанавливается прослеживаемость проблем и возможностей бизнеса или цели работы и предпочтительных классов альтернативных решений.

6.4.1.3 Деятельность и задачи

Проект должен реализовать следующие виды деятельности и задачи в соответствии с применимыми политиками и процедурами организации в отношении процесса анализа бизнеса.

a) Подготовка к анализу бизнеса или цели работы. Деятельность состоит из следующих задач:

1) Анализ выявленных проблем и возможностей в стратегии организации в отношении желаемых целей или задач организации.

Примечание – Сюда входят проблемы или возможности в отношении бизнеса или цели работы организации, видения, концепции деятельности и других стратегических целей и задач организации. Это включает выявленные недостатки или пробелы в существующих возможностях, системах, продуктах или услугах.

2) Определение стратегии анализа бизнеса или цели работы.

Примечание – Включает подход, который будет использоваться для идентификации и определения проблемного пространства, характеризации пространства решений и выбора класса решений.

3) Определение и планирование необходимых вспомогательных систем или услуг, необходимых для поддержки анализа бизнеса или цели работы.

Примечание – Включает определение требований и интерфейсов для вспомогательных систем. К вспомогательным системам для анализа бизнеса или цели работы относятся бизнес-системы и хранилища организации или других доступных субъектов.

4) Получение или приобретение доступа к используемым системам или услугам, обеспечивающим возможность их использования.

Примечание – Процесс валидации используется для объективного подтверждения того, что вспомогательная система достигает целевого использования для своих вспомогательных функций.

b) Определение проблемы или пространства возможностей. Деятельность состоит из следующих задач:

1) Анализ жалоб, проблем и возможностей клиентов в контексте соответствующих факторов торгового пространства.

Примечание

1 Анализ сосредоточен на понимании масштаба, основы или движущих сил проблем или возможностей, в отличие от синтеза, который является фокусом системного анализа и управления решениями, необходимых для торговых исследований. В фокус внимания здесь входят изменения в требованиях миссии, бизнес-возможности, возможности, улучшение производительности или недостаток существующих систем, улучшение безопасности и защиты, такие факторы, как стоимость и эффективность, изменения в регулировании, неудовлетворенность пользователей, а также факторы PESTEL (политические, экономические, социальные, технологические, экологические и правовые). Соответствующие факторы могут быть определены с помощью внешнего, внутреннего или SWOT-анализа (Сильные стороны, слабые стороны, возможности и угрозы).

2 Результаты анализа рассматриваются как часть решений по управлению проектным портфелем.

2) Определение цели работы, бизнеса или операционной проблемы или возможности.

Примечание – Определение включает контекст и любые ключевые параметры, независимо от конкретного решения, поскольку решение может быть операционным изменением, изменением существующего продукта или услуги или новой системой.

c) Определение пространства решений. Деятельность состоит из следующих задач:

1) Определение предварительных операционных концепций и других концепций на стадиях жизненного цикла.

Примечания

1 Включает определение основных групп заинтересованных сторон, таких как клиенты, пользователи, администрации, регулирующие органы и владельцы систем, которые определены в процессе определения потребностей и требований заинтересованных сторон.

2 Предварительные концепции жизненного цикла включают предварительные концепции приобретения, предварительные концепции развертывания, предварительные рабочие концепции, предварительные концепции поддержки и предварительные концепции вывода из эксплуатации. Эксплуатационные концепции включают в себя режимы или состояния работы высокого уровня, рабочие сценарии, потенциальные варианты использования или использование в рамках предлагаемой бизнес-стратегии. Концепции позволяют проводить анализ осуществимости и оценку альтернатив. Концепции дополнительно уточняются в процессе определения потребностей и требований заинтересованных сторон.

3 Операционная среда может иметь известные уязвимости, связанные с конкретными угрозами безопасности и опасностями безопасности. Уязвимости необходимо понимать в связи с разрабатываемым продуктом. Системный и человеческий интерфейсы являются элементом контекста системного обеспечения, и связанные с ними уязвимости исследуются в контексте критически важных угроз.

2) Определение возможных альтернативных классов решений, которые охватывают потенциальное пространство решений.

Примечание – Классы могут варьироваться от простых операционных изменений до различных разработок или модификаций программных систем. Пространство решений может включать идентификацию существующих активов, систем и программных продуктов, пригодных для повторного использования, а также изменения в услугах, которые могут удовлетворить потребность в операционных или функциональных изменениях. Это включает вывод о том, какие потенциальные ожидаемые услуги будут необходимы. Характеристика пространства решений часто использует процесс определения архитектуры для определения точки зрения пользовательской архитектуры, в результате чего формируются представления архитектуры (например, представления возможностей, представления программы и операционные представления), как предложено в стандарте [ISO/IEC/IEEE 42010].

d) Оценка альтернативных классов решений. Деятельность состоит из следующих задач:

1) Оценка каждого класса альтернативных решений.

Примечания

1 Каждый класс альтернативных решений оценивается по определенным критериям, которые устанавливаются на основе стратегии организации. Целесообразность класса решений является одним из ключевых критериев принятия решения. Процесс управления проектным портфелем предоставляет некоторые критерии для рассмотрения.

2 Процесс системного анализа используется для оценки значения каждого критерия для каждого класса альтернативных решений. Рекомендуется структурировать компромиссы по доступности цены. Включение стоимости в качестве критерия будет способствовать принятию решений о доступности. Оценка альтернатив может включать моделирование, имитацию, аналитические методы или экспертную оценку для понимания рисков, осуществимости и ценности альтернативных классов решений-кандидатов.

2) Выбор предпочтительного(ых) класса(сов) альтернативных решений.

Примечание – Процесс управления принятием решений используется для оценки альтернатив и руководства выбором. Выбранные альтернативы проверяются в контексте стратегии организации. Обратная связь по рискам, осуществимости, рыночным факторам и альтернативам предоставляется для использования в обновлении стратегии организации.

e) Управление анализом бизнеса или целью работы. Деятельность состоит из следующих задач:

1) Поддержка прослеживаемости анализа бизнеса или цели работы.

Примечание – На протяжении всего жизненного цикла поддерживается двунаправленная прослеживаемость между проблемами и возможностями бизнеса и цели работы и предпочтительными альтернативными классами решений с организационной стратегией, потребностями и требованиями заинтересованных сторон, а также результатами системного анализа, поддерживающими решения.

2) Предоставление ключевых аспектов и информационных элементов, которые были выбраны для базовых линий.

Примечание – Процесс управления конфигурацией используется для создания и поддержания элементов конфигурации и базовых линий. Этот процесс определяет кандидатов для базовой линии, а процесс управления информацией контролирует информационные элементы.

6.4.2 Процесс определения потребностей заинтересованных сторон

6.4.2.1 Цель

Целью процесса определения потребностей и требований заинтересованных сторон является определение требований заинтересованных сторон к системе, которая может обеспечить возможности, необходимые пользователям и другим заинтересованным сторонам в определенной среде.

Процесс определяет заинтересованные стороны или классы заинтересованных сторон, вовлеченные в работу с системой на протяжении всего ее жизненного цикла, и их потребности. Он анализирует и преобразует эти потребности в общий набор требований заинтересованных сторон, которые выражают предполагаемое взаимодействие системы с операционной средой и являются эталоном, по которому проверяется каждая полученная операционная способность. Требования заинтересованных сторон определяются с учетом контекста интересующей системы с взаимодействующими системами и вспомогательными системами.

Примечание – В области знаний [SWEBOK, Guide to the Software Engineering Body of Knowledge, Software Requirements] обсуждаются основы требований к программному обеспечению (например, определение, типы, свойства, характеристики качества) и другие темы, такие как заинтересованные стороны, выявление требований, анализ и управление, которые обеспечивают дополнительное руководство для программных систем.

6.4.2.2 Результаты

В результате успешного внедрения процесса определения потребностей и требований заинтересованных сторон:

a) Определяются заинтересованные стороны системы.

b) Определяются требуемые характеристики и контекст использования возможностей и концепций на этапах жизненного цикла, включая оперативные концепции.

c) Определяются ограничения на систему.

d) Определяются потребности заинтересованных сторон.

e) Потребности заинтересованных сторон определяются по приоритетам и преобразуются в четко сформулированные требования заинтересованных сторон.

f) Определяются критические показатели эффективности.

g) Достигается согласие заинтересованных сторон с тем, что их потребности и ожидания адекватно отражены в требованиях.

h) Доступны любые вспомогательные системы или услуги, необходимые для удовлетворения потребностей и требований заинтересованных сторон.

i) Устанавливается связь требований заинтересованных сторон с заинтересованными сторонами и их потребностями.

6.4.2.3 Деятельность и задачи

Проект должен выполнить следующие мероприятия и задачи в соответствии с применимыми политиками и процедурами организации в отношении процесса определения потребностей и требований заинтересованных сторон.

a) Подготовка к определению потребностей заинтересованных сторон и требований. Деятельность состоит из следующих задач:

1) Определение заинтересованных сторон, которые имеют интерес к программной системе на протяжении всего ее жизненного цикла.

Примечание – Сюда входят лица и классы заинтересованных сторон, которые являются пользователями, операторами, сторонниками, разработчиками, производителями, инструкторами, эксплуатационщиками, распорядителями, организациями-покупателями и поставщиками, сторонами, ответственными за внешние сопрягающие структуры, регулирующими органами и другими лицами, имеющими законный интерес к системе. Там, где прямая связь нецелесообразна (например, для потребительских товаров и услуг), выбираются представители или назначенные доверенные лица заинтересованных сторон.

2) Определение потребностей заинтересованных сторон и стратегии определения требований.

Примечание – Некоторые заинтересованные стороны имеют интересы, которые противоречат интересам приобретателя (например, конкуренты на рынке, хакеры, террористы) или противостоят друг другу. Когда интересы заинтересованных сторон противостоят друг другу, но не противостоят программной системе, этот процесс предназначен для достижения согласия между классами заинтересованных сторон для установления общего набора приемлемых требований. Намерения или желания тех, кто противостоит приобретателям, или недоброжелателей системы, учитываются в процессе управления рисками, анализа угроз в процессе системного анализа или требований к системе/программному обеспечению по безопасности, адаптивности или устойчивости. В этом случае потребности заинтересованных сторон не удовлетворяются, а решаются таким образом, чтобы помочь обеспечить надежность и целостность системы в случае возникновения действий со стороны недоброжелателей.

3) Определение и планирование необходимых вспомогательных систем или услуг, необходимых для поддержки потребностей заинтересованных сторон и определения требований.

Примечание – Получение или приобретение доступа к используемым системам или услугам, обеспечивающим возможность их использования.

4) Получение или приобретение доступа к используемым системам или услугам, обеспечивающим возможность их использования.

Примечание – Процесс валидации используется для объективного подтверждения того, что вспомогательная система достигает целевого использования для своих вспомогательных функций.

b) Определение потребности заинтересованных сторон. Деятельность состоит из следующих задач:

1) Определение контекста использования в рамках концепции операций и концепций предварительного жизненного цикла.

Примечание – Контекст использования часто фиксируется с помощью описания контекста использования [ISO/IEC 25063]. Предварительные концепции жизненного цикла разрабатываются в процессе анализа бизнеса или цели работы.

2) Определение потребности заинтересованных сторон.

Примечания

1 Определение потребностей заинтересованных сторон включает в себя выяснение потребностей непосредственно у заинтересованных сторон, выявление неявных потребностей заинтересованных сторон на основе знания области и понимания контекста, а также документированных пробелов в результате предыдущей деятельности. Потребности часто включают показатели эффективности. Для помощи в выявлении потребностей часто используется функциональный анализ. Характеристики качества модели качества в [ISO/IEC 25010] и применение модели качества к анализу требований в [ISO/IEC 25030] полезны для выявления и идентификации требований к качеству нефункциональных требований, которые часто являются неявными потребностями заинтересованных сторон.

2 В области знаний [SWEBOK] обсуждаются некоторые дополнительные методы для выявления и уточнения требований к программному обеспечению, такие как прототипирование, наблюдение, пользовательские истории для определения необходимой функциональности, поиск данных и анализ продуктов конкурентов.

3 Потребности заинтересованных сторон описывают потребности, желания, пожелания, ожидания и воспринимаемые ограничения идентифицированных заинтересованных сторон. Понимание потребностей заинтересованных сторон в отношении минимальных требований к безопасности и конфиденциальности, необходимых для операционной среды, сводит к минимуму возможность нарушения планов, графиков и производительности. Если существует вероятность возникновения значительных проблем, связанных с пользователями и другими заинтересованными сторонами и их участием в программной системе или взаимодействием с ней, рекомендации по выявлению и устранению проблем «человек-система» представлено в [ISO TS 18152].

3) Расстановка приоритетов и сокращение отбора потребностей.

Примечание – Процесс управления принятием решений обычно используется для поддержки определения приоритетов. Процесс системного анализа используется для анализа потребностей на предмет осуществимости или других факторов.

4) Определение потребности заинтересованных сторон и их обоснование.

Примечание – Потребности концентрируются на назначении и поведении системы и описываются в контексте операционной среды и условий. Целесообразно проследить потребности до их ресурсов и обоснования.

c) Разработка операционной концепции и других концепций жизненного цикла. Деятельность состоит из следующих задач:

Примечание – Другие концепции жизненного цикла могут включать концепции приобретения, развертывания, поддержки, безопасности и списания. В этой деятельности предварительные концепции жизненного цикла, определенные в рамках процесса анализа бизнеса или миссии, получают дальнейшее развитие в контексте конкретных потребностей заинтересованных сторон, поскольку определяются соответствующие сценарии и взаимодействия. Дополнительная информация об операционных концепциях приведена в п. 5 и 6 [ISO/IEC/IEEE 29148:2011], а аннотированную схему операционной концепции системы – в приложении A [ISO/IEC/IEEE 29148:2011].

1) Определение репрезентативного набора сценариев для выявления требуемых возможностей, соответствующих предполагаемым операционным и другим концепциям жизненного цикла.

Примечания

1 Сценарии используются для анализа работы системы в предполагаемой среде с целью выявления дополнительных потребностей или требований, которые, возможно, не были явно определены ни одной из заинтересованных сторон, например, юридические, нормативные и социальные обязательства. Контекст использования системы определяется и анализируется, включая действия, которые пользователи выполняют для достижения целей системы, соответствующие характеристики пользователей (например, ожидаемая подготовка и знания, частота использования системы, обязанности, проблемы доступности), физическая среда (например, доступное освещение, температура) и любое оборудование, которое будет использоваться (например, защитное или коммуникационное оборудование). При необходимости анализируется социальное и организационное влияние на пользователей, которое влияет на использование системы или ограничивает ее дизайн. Сценарии, сосредоточенные на злоумышленниках, их среде, инструментах, методах и возможностях, являются ключевыми соображениями при разработке оперативной концепции. Сценарии расставляются по приоритетам, чтобы отразить взвешенную важность различных оперативных потребностей.

2 Сценарии часто мотивируют обновление эксплуатационных или других концепций жизненного цикла. Сценарии злоупотреблений и отказов подчеркивают необходимость дополнительных функциональных требований (или более конкретных производных требований) для снижения рисков, которые определены в сценариях злоупотреблений или отказов.

2) Определение факторов, влияющих на взаимодействие между пользователями и системой.

i) Предполагаемые физические, умственные и познавательные способности пользователей;

ii) Рабочее место, окружающая среда и оборудование, включая другое оборудование в контексте использования;

iii) нормальные, необычные и аварийные условия;

iv) Набор, обучение и культура операторов и пользователей.

Примечания

1 Требования к удобству использования учитывают возможности человека и ограничения его навыков. По возможности используются применимые стандарты, например [ISO 9241], и принятые профессиональные практики.

2 Если удобство использования важно, требования к удобству использования планируются, определяются и реализуются в рамках процессов жизненного цикла. Информация о проблемах пользователей системы приведена в [ISO TS 18152], информация об удобстве использования – [ISO/IEC 25060:2010].

d) Преобразование потребностей заинтересованных сторон в требования заинтересованных сторон. Деятельность состоит из следующих задач:

1) Определение ограничений на системное решение.

Примечание – Ограничения могут быть результатом 1) случаев или областей решения, определенных заинтересованными сторонами; 2) решений по реализации, принятых на более высоких уровнях иерархической структуры системы; 3) требуемого использования определенных вспомогательных, унаследованных или сопряженных систем, элементов системы, ресурсов и персонала; или 4) определенных заинтересованными сторонами целей доступности. Включая те, которые являются неизбежными последствиями существующих соглашений, решений руководства и технических решений.

2) Определение требований и функций заинтересованных сторон, относящиеся к критическим характеристикам качества, таким как обеспечение, безопасность, охрана, окружающая среда или здоровье.

Примечания

1 Дополнительная информация об обеспечении систем и программного обеспечения приведена в [ISO/IEC/IEEE 15026].

2 Выявление рисков безопасности облегчает определение требований и функций безопасности. Риски безопасности включают риски, связанные с методами работы и поддержки, здоровьем и безопасностью, угрозами имуществу и влиянием окружающей среды. Необходимо применять стандарты и общепринятую профессиональную практику. Например, [IEC 61508:2010] содержит подробные требования.

3 Выявление рисков безопасности облегчает определение дополнительных требований и функций безопасности. При необходимости включаются применимые области безопасности системы, такие как физическая, процедурная, коммуникационная, компьютеры, программы, данные и выбросы. Это включает доступ и ущерб находящемуся под защитой персоналу, имуществуи информации, компрометацию конфиденциальной информации и отказ в разрешенном доступе к имуществу и информации. Сюда также входят необходимые функции безопасности, такие как смягчение и локализация, со ссылкой на применимые стандарты и принятую профессиональную практику, где это обязательно или уместно. Обзор жизненного цикла безопасности программного обеспечения приведен в Приложении E (E.6).

4 Дополнительная информация о характеристиках качества с точки зрения использования приведена в [ISO/IEC 25030].

3) Определение требований заинтересованных сторон в соответствии с концепциями жизненного цикла, сценариями, взаимодействиями, ограничениями и критическими характеристиками качества.

Примечания

1 Дополнительная информация о требованиях заинтересованных сторон приведена в [ISO/IEC/IEEE 29148:2011 Пункты 5 и 6], описание и аннотированный набросок спецификации требований заинтересованных сторон – [ISO/IEC/IEEE 29148:2011 пункты 8 и 9].

2 Требования заинтересованных сторон пересматриваются на ключевых этапах жизненного цикла, чтобы обеспечить учет изменений в потребностях.

3 Требования заинтересованных сторон записываются в форме, пригодной для управления требованиями на протяжении всего жизненного цикла. Записи устанавливают базовую линию требований заинтересованных сторон и сохраняют изменения потребностей и их происхождение на протяжении всего жизненного цикла программного обеспечения. Записи являются основой для отслеживания решений, принятых в процессе анализа бизнеса или миссии, а также потребностей заинтересованных сторон, системных требований и последующих элементов программной системы.

4 Требования заинтересованных сторон являются основой критериев валидации программной системы и ее элементов.

e) Анализ требований заинтересованных сторон. Деятельность состоит из следующих задач:

1) Анализ полного набора требований заинтересованных сторон.

Примечания

1 Требования заинтересованных сторон анализируются на предмет характеристик отдельных требований, а также характеристик набора требований. К потенциальным характеристикам анализа относится то, что требования являются необходимыми, свободными от реализации, однозначными, последовательными, полными, единичными, выполнимыми, прослеживаемыми, проверяемыми, доступными и ограниченными. Дополнительная информация о характеристиках требований приведена в [ISO/IEC/IEEE 29148].

2 Процесс системного анализа используется для оценки осуществимости и экономической целесообразности. Процессы верификации и валидации используются при рассмотрении требований заинтересованных сторон.

2) Определение критических показателей эффективности, позволяющих оценить технические достижения.

Примечание – Это включает определение технических и качественных показателей и критических параметров эффективности, связанных с каждым показателем эффективности, определенным в требованиях заинтересованных сторон. Критические показатели эффективности (например, меры эффективности и меры пригодности) определяются, анализируются и пересматриваются, чтобы помочь обеспечить выполнение требований заинтересованных сторон и помочь обеспечить идентификацию риска стоимости, графика или производительности проекта, связанного с любым несоответствием. В [ISO/IEC 15939] приводится процесс для выявления, определения и использования соответствующих мер. В [INCOSE TP-2003-020-01] приводится определении и внедрении критических показателей эффективности. Серия стандартов [ISO/IEC 25000] содержит соответствующие показатели качества.

3) Передача проанализированных требований соответствующим заинтересованным сторонам, чтобы убедиться, что их потребности и ожидания были адекватно отражены и выражены.

4) Решение вопросов, связанных с требованиями заинтересованных сторон.

Примечание – Сюда входят требования, которые нарушают характеристики для отдельных требований или набора требований, как определено в [ISO/IEC/IEEE 29148].

f) Управление потребностями заинтересованных сторон и определение требований. Эта деятельность состоит из следующих задач:

1) Получение четкого согласия с определенными заинтересованными сторонами в отношении требований заинтересованных сторон.

Примечание – Включает подтверждение того, что требования заинтересованных сторон выражены правильно, понятны, и что разрешение конфликта в требованиях не испортило и не поставило под угрозу намерения заинтересованных сторон.

2) Поддержание прослеживаемости потребностей и требований заинтересованных сторон.

Примечание – На протяжении всего жизненного цикла поддерживается двунаправленная прослеживаемость между потребностями и требованиями заинтересованных сторон, стратегией организации, проблемами и возможностями бизнеса и миссии. Требования заинтересованных сторон прослеживаются до требований к системе/программному обеспечению в процессе определения требований к системе/программному обеспечению. Прослеживаемость часто поддерживается с помощью соответствующего хранилища данных.

3) Предоставление ключевых аспектов и информационных элементов, которые были выбраны для базовых линий.

Примечание – Процесс управления конфигурацией используется для создания и поддержания элементов конфигурации и базовых линий. Процесс определяет кандидатов для базовой линии, а процесс управления информацией контролирует информационные элементы.

Для этого процесса потребности заинтересованных сторон, требования заинтересованных сторон и операционная концепция являются типичными информационными элементами, которые определяются на базовой основе.

6.4.3 Процесс определения требований к системе/программному обеспечению

6.4.3.1 Цель

Цель процесса определения требований к системе/программному обеспечению - преобразовать представление заинтересованных сторон, ориентированное на пользователя, о желаемых возможностях в техническое представление решения, которое отвечает оперативным потребностям пользователя.

процесс создает набор измеримых системных требований, которые определяют, с точки зрения поставщика, какими характеристиками, атрибутами, функциональными и эксплуатационными требованиями должна обладать система, чтобы удовлетворить требования заинтересованных сторон. Насколько позволяют ограничения, требования не должны подразумевать какую-либо конкретную реализацию.

Примечания

1 При высокоуровневом представлении программной системы этот процесс может быть использован для определения общих требований к системе. По мере декомпозиции программной системы на элементы, каждый элемент, в свою очередь, рассматривается как система, функция или набор функций, и этот процесс может быть использован для дальнейшего определения требований. Анализ требований и инструменты поддерживают прослеживаемость требований между программной системой и ее элементами.

2 В области знаний [SWEBOK] обсуждается определение, анализ, моделирование, спецификация, валидация, управление и другие темы, которые служат дополнительным руководством для программных систем..

3 Формулировка итогов процесса определения требований к системе/программному обеспечению несколько отличается от итогов процесса определения требований к системе в [ISO/IEC/IEEE 15288:2015]. Использование формулировки «системные/программные требования» подчеркивает применимость данного документа к программным системам, имеющим программные требования и системные требования. Это призвано помочь пользователям, которые определяют системные требования и требования к программному обеспечению иерархически или на разных этапах.

6.4.3.2 Результаты

В результате успешного внедрения процесса определения требований к системе/программному обеспечению:

a) Определяется описание системы или элемента, включая интерфейсы, функции и границы, для системного решения.

b) Определяются требования к системе/программному обеспечению (функциональные, эксплуатационные, технологические, нефункциональные и интерфейсные) и ограничения на проектирование.

c) Определяются критические показатели эффективности.

d) Анализируются требования к системе/программному обеспечению.

e) Доступны любые вспомогательные системы или услуги, необходимые для определения требований к системе/программному обеспечению.

f) Прослеживаются требования к системе/программному обеспечению до требований заинтересованных сторон.

6.4.3.3 Деятельность и задачи

Проект должен реализовать следующие мероприятия и задачи в соответствии с применимыми политиками и процедурами организации в отношении процесса определения требований к системе/программному обеспечению.

a) Подготовка к определению требований к системе/программному обеспечению. Деятельность состоит из следующих задач:

1) Определение функциональных границ программной системы или элемента с точки зрения поведения и предоставленных свойств.

Примечание – Определение функциональных границ частично основано на контексте использования и операционных сценариях, определенных в рамках процесса определения потребностей и требований заинтересованных сторон. Сюда входят стимулы (входные данные) программной системы и ее реакция на действия пользователей и внешних систем, а также анализ и описание требуемых взаимодействий между программной системой и ее операционной средой в терминах свойств и ограничений интерфейса, таких как процедурные потоки, порядок вызова, форматы и потоки данных, пропускная способность и время. Это устанавливает ожидаемое поведение программной системы, выраженное в количественных терминах, на ее границе. Для программного обеспечения границы обычно выражаются в интерфейсах прикладных программ (API) и графическом интерфейсе пользователя (GUI) или интерфейсных файлах или службах, включая форматы данных. Приложение E (E.5) предоставляет представление управления интерфейсами в процессах жизненного цикла.

2) Определение стратегии определения требований к системе/программному обеспечению.

Примечание – Включает подход, который будет использоваться для выявления, определения и управления требованиями к системе/программному обеспечению с выбранной моделью жизненного цикла, например, эволюционной, инкрементной или итерационной. На стратегию могут влиять многие факторы, например, сложность программной системы и информации и функций, которыми необходимо управлять; необходимость обеспечения свободного доступа и общего понимания несколькими членами команды; степень совместного участия приобретателя или представителей пользователей на всех этапах разработки; предполагает ли проект новую разработку, модификацию, повторное использование или интеграцию существующих систем; требования к документации процесса, включая период хранения. Модель жизненного цикла будет влиять на то, когда и как часто будет проводиться определение требований к системе/программному обеспечению. Приложение H описывает постепенную разработку требований в проектах, использующих гибкие методы.

3) Определение и планирование необходимых вспомогательных систем или услуг, необходимых для поддержки определения требований к системе/программному обеспечению.

Примечание – Включает определение требований и интерфейсов для вспомогательных систем. К вспомогательным системам для определения требований относятся инструменты для содействия и управления требованиями. Инструменты для управления требованиями к программному обеспечению, интегрированные с разработкой, тестированием и СМ, могут упростить отслеживание и ускорить создание программного обеспечения. Планы для вспомогательных систем и описание методов моделирования, используемых для поддержки процесса определения требований к системе/программному обеспечению, могут быть включены в SDP.

4) Получение или приобретение доступа к используемым системам или услугам, обеспечивающим возможность их использования.

Примечание – Процесс валидации используется для объективного подтверждения того, что вспомогательная система достигает целевого использования для своих вспомогательных функций.

b) Определение требований к системе/программному обеспечению. Деятельность состоит из следующих задач:

1) Определение каждой функции, которую должна выполнять система или элемент программного обеспечения.

Примечания

1 Функции программного обеспечения могут быть описаны в сценариях использования, пользовательских историях или сценариях и предполагают преобразование данных и информации для достижения потребностей пользователей (требований заинтересованных сторон). В некоторых случаях функции определяются на основе анализа критических характеристик качества, таких как производительность, безопасность или доступность (например, функция диагностики системы или функция резервного копирования данных с высокой частотой для обеспечения надежности).

2 Вспомогательные функции, которые необходимы для поддержки интересующей системы в достижении ее функциональности, также идентифицируются и определяются одновременно с функцией интересующей системы. Это необходимо для того, чтобы обеспечить идентификацию и учет вспомогательных функций в системной среде.

2) Определение требуемого состояния или режима работы программной системы.

Примечания

1 Состояния или режимы работы могут быть смоделированы и представлены в нескольких методах моделирования и перспективах, чтобы дать достаточно полное описание желаемых требований к системе или элементу.

2 Условия выполнения функций часто предполагают взаимодействие между функциями или элементами. Например, некоторые требования к программному обеспечению (например, ограничение по времени выполнения) могут быть распределены между несколькими элементами программной системы, что влияет на обработку требования в тестовом примере или регрессионном тесте.

3) Определение необходимого ограничения реализации.

Примечание – Для элементов программного обеспечения это включает решения по реализации, которые выделяются из определения архитектуры на более высоких уровнях в программной системе, вводятся требованиями заинтересованных сторон или ограничениями решения. Ограничения реализации включают условия, при которых система должна быть способна выполнять функцию, условия, при которых система должна начать выполнение этой функции (вход) и условия, при которых система должна прекратить выполнение этой функции (выход).

4) Определение требования, которые относятся к рискам, критичности программной системы или критическим характеристикам качества.

Примечания

1 Нефункциональные требования и критические характеристики качества в программных системах обычно включают требования, связанные со здоровьем, безопасностью, безопасностью и гарантией программного обеспечения, надежностью, доступностью и поддерживаемостью (ремонтопригодностью), а также временными ограничениями для пропускной способности и производительности. Процесс системного анализа может быть использован для определения соответствующих значений требований к производительности с учетом предполагаемых затрат на их достижение и их влияния на эксплуатацию и использование системы.

2 Анализ и определение критериев безопасности включают требования, относящиеся к методам эксплуатации и технического обслуживания, воздействию окружающей среды и риску травмирования персонала. Он также включает выражение каждой функции, связанной с безопасностью, и связанной с ней целостности, с точки зрения необходимого снижения риска и распределения по назначенным системам, связанным с безопасностью. Используются применимые стандарты, касающиеся функциональной безопасности, например, [IEC 61508], и охраны окружающей среды, например, [ISO 14001]. Анализ включает соображения безопасности, например, связанные с компрометацией и защитой конфиденциальной информации, данных и материалов. Определяются риски, связанные с безопасностью, включая административные, кадровые, физические, компьютерные, коммуникационные, сетевые, эмиссионные и экологические факторы с использованием, при необходимости, применимых стандартов безопасности. Руководство по обеспечению безопасности систем и программного обеспечения приведено в [ISO/IEC/IEEE 15026-4, ISO/IEC 27036] содержит руководство по требованиям информационной безопасности при аутсорсинге продуктов и услуг. [ISO 25030] содержит руководство по факторам и характеристикам качества внешней системы. Приложение E (E.6) описывает представление обеспечения качества программного обеспечения в процессах жизненного цикла.

3 Для программных систем, предназначенных для взаимодействия с человеком, рассматриваются спецификации по проектированию человеческого фактора (эргономика). Для систем, к которым предъявляются требования по удобству использования, рекомендации по получению желаемого уровня удобства использования можно найти в [ISO/FDIS 9241-220].

5) Определение требования к системе/программному обеспечению и атрибутов требований, включая следующее:

i) Элементы данных, структуры и форматы данных, а также требования к хранению базы данных или данных;

ii) Пользовательские интерфейсы и пользовательская документация (информация для пользователей) и обучение пользователей;

iii) Интерфейсы с другими системами и услугами;

iv) Функции и нефункциональные характеристики, включая критические характеристики качества и целевые показатели затрат;

v) Переход операционных процессов и данных из существующих автоматизированных и ручных систем, подход и график миграции, установка программного обеспечения и приемка продукта; и

vi) Атрибуты требований, такие как обоснование; приоритет; прослеживаемость до элементов программной системы, тестовых примеров и информационных элементов; методы проверки; включение в утвержденные базовые линии; и оцененный риск.

Примечания

1 Определение требований включает итеративные и рекурсивные шаги параллельно с другими процессами жизненного цикла. В зависимости от используемой модели жизненного цикла необходимо сравнить ресурсы, которые затрачиваются на обеспечение первоначальной корректности требований, с ресурсами, необходимыми для развития требований на основе результатов верификации и валидации.

2 Требования и атрибуты системы/программного обеспечения записаны с уровнем детализации и в форме, подходящей для управления требованиями на протяжении всего жизненного цикла. В [ISO/IEC/IEEE 29148:2011 Пункты 5 и 6] приведена дополнительная информация о требованиях, а также в пунктах 8 и 9 – об описании и аннотированной схеме спецификации требований к системе и спецификации требований к программному обеспечению.

c) Анализ требований к системе/программному обеспечению. Деятельность состоит из следующих задач:

1) Анализ полного набора требований к системе/программному обеспечению.

Примечания

1 Требования анализируются на предмет характеристик отдельных требований, а также характеристик совокупности требований. К потенциальным характеристикам анализа относится то, что требования являются необходимыми, не требующими реализации, однозначными, последовательными, полными, единичными, выполнимыми, прослеживаемыми, проверяемыми, доступными и ограниченными. Процесс верификации используется для определения того, соответствуют ли требования атрибутам и характеристикам хороших требований. В некоторых случаях оценивается техническая и экономическая целесообразность валидации и верификации альтернативных формулировок требований. [ISO/IEC/IEEE 29148] предоставляет дополнительную информацию о характеристиках требований.

2 Процесс системного анализа может быть использован для оценки осуществимости, доступности, сбалансированности и других характеристик требований. Процесс системного анализа используется для определения соответствующих значений параметров требований с учетом предполагаемой стоимости, графика и технических характеристик программной системы.

3 Можно определить приоритетность требований исходя из того, что некоторые требования могут быть выполнены постепенно или даже отложены или отменены.

2) Определение критических показателей эффективности, позволяющих оценить технические достижения.

Примечание – Включает определение технических мер и мер качества и критических параметров эффективности, связанных с каждой мерой эффективности, определенной в требованиях к элементам программной системы. Критические показатели эффективности (например, меры эффективности и технические меры эффективности) анализируются и пересматриваются, чтобы помочь обеспечить выполнение требований к системе/программному обеспечению и идентификацию риска стоимости проекта, графика или производительности, связанного с любым несоответствием. [ISO/IEC 15939] предоставляет процесс для выявления, определения и использования соответствующих мер. [INCOSE TP-2003-020-01] предоставляет информацию о выборе, определении и внедрении критических мер производительности. Серия стандартов [ISO/IEC 2500]0 содержит соответствующие меры.

3) Передача проанализированных требований соответствующим заинтересованным сторонам для рассмотрения.

Примечание – Обратная связь помогает подтвердить, что указанные требования были адекватно зафиксированы и выражены. Подтверждается, что они являются необходимым и достаточным ответом на требования заинтересованных сторон и необходимым и достаточным вкладом в другие процессы, в частности, архитектуру, проектирование и проверку программного обеспечения. Процесс валидации используется для определения того, соответствуют ли требования к системе/программному обеспечению потребностям пользователей.

4) Выявление и решение проблем, недостатков, конфликтов и слабых мест в рамках полного набора требований.

Примечание – Сюда относятся требования, которые не поддаются проверке, неоднозначны, нарушают характеристики для отдельных требований или не согласуются с другими в наборе требований. Решение проблем с требованиями может быть итерационным в рамках определенных моделей жизненного цикла.

d) Управление требованиями к системе/программному обеспечению. Деятельность состоит из следующих задач:

Примечание – Поддержание требований к системе/программному обеспечению включает определение, регистрацию и контроль базовой линии, обычно в рамках формального управления конфигурацией, а также управление изменениями, возникающими в результате применения других процессов жизненного цикла, таких как архитектура или проектирование.

1) Получение подтверждения по требованиям к системе/программному обеспечению.

Примечание – Включает подтверждение того, что требования к системе/программному обеспечению выражены правильно, понятны создателям и исполнителям, и что разрешение конфликта в требованиях соответствует решениям заинтересованных сторон.

2) Поддержка прослеживаемости требований к системе/программному обеспечению.

Примечание – На протяжении всего жизненного цикла поддерживается двунаправленная прослеживаемость между требованиями к системе/программному обеспечению и требованиями заинтересованных сторон, архитектурными объектами, определениями интерфейсов, результатами анализа, методами или методикам проверки, а также распределенными, декомпозированными и производными требованиями. Прослеживаемость позволяет верифицировать, что достижимые требования заинтересованных сторон удовлетворяются одним или несколькими требованиями к системе или элементам, и такие требования удовлетворяют или способствуют выполнению не менее одного требования заинтересованной стороны. Прослеживаемость часто обеспечивается соответствующим репозиторием данных или интегрированной инфраструктурой разработки и тестирования.

3) Предоставление ключевых аспектов и информационных элементов, которые были выбраны для базовых линий.

Примечание – Процесс управления конфигурацией используется для создания и поддержки элементов конфигурации и базовых показателей. Процесс определяет кандидатов для базового плана, а процесс управления информацией контролирует информационные элементы, такие как спецификации требований. Для этого процесса требования к системе/программному обеспечению являются типичными артефактами, которые определены как базовые.

6.4.4 Процесс определения архитектуры

6.4.4.1 Цель

Целью процесса определения архитектуры является генерация альтернатив архитектуры системы, выбор одной или нескольких альтернатив, которые отражают интересы заинтересованных сторон и отвечают требованиям системы, и выражение этого в наборе согласованных представлений.

Итерация процесса определения архитектуры с процессом анализа бизнеса или миссии, процессом определения требований к системе/программному обеспечению, процессом определения дизайна и процессом определения потребностей и требований заинтересованных сторон используется для того, чтобы достичь согласованного понимания проблемы, которую необходимо решить, и определить удовлетворительное решение. Результаты процесса определения архитектуры используются во всех процессах жизненного цикла. Определение архитектуры может применяться на многих уровнях абстракции, выделяя соответствующие детали, которые необходимы для принятия решений на данном уровне.

Примечания

1 Архитектура системы имеет дело с фундаментальными принципами, концепциями, свойствами и характеристиками и их включением в интересующую систему. Определение архитектуры имеет больше применений, чем просто драйвер или часть проектирования. В [ISO/IEC/IEEE 42010:2011] приведена дополнительная информация об описании архитектуры, а также об использовании и природе архитектуры.

2 Процесс определения архитектуры поддерживает идентификацию заинтересованных сторон и их проблем. По мере того, как процесс разворачивается, появляется понимание связи между требованиями, указанными для программной системы, и возникающими свойствами и поведением системы, которые возникают в результате взаимодействия и отношений между элементами системы. Эффективная архитектура максимально диагностична по дизайну, чтобы обеспечить максимальную гибкость в пространстве проектных решений. Даже для программной системы, состоящей из одного продукта, дизайн продукта, скорее всего, будет меняться с течением времени, в то время как архитектура остается неизменной. Эффективная архитектура также выделяет и поддерживает компромиссы для процесса определения дизайна и, возможно, других процессов, таких как, управление портфелем, планирование проекта, определение требований к системе/программному обеспечению и верификация.

3 Определение архитектуры может применяться к линейке продуктов, а не к одной программной системе. Архитектура линейки продуктов описывает структурные свойства для построения группы связанных систем с общими компонентами и взаимосвязями. В архитектуре линейки продуктов архитектура обязательно охватывает несколько проектов. Архитектура служит для придания линейке продуктов целостности и помогает обеспечить совместимость и взаимодействие в рамках всей линейки продуктов. [ISO/IEC 26550:2013] описывает создание архитектуры домена для линейки продуктов.

4 В областях знаний [SWEBOK] обсуждаются ключевые аспекты архитектуры программного обеспечения во взаимосвязи с системой, а также в отношении итераций с проектированием.

6.4.4.2 Результаты

В результате успешного внедрения процесса определения архитектуры:

a) Выявленные проблемы заинтересованных сторон учитываются в архитектуре.

b) Разрабатываются точки зрения на архитектуру.

c) Определяются контекст, границы и внешние интерфейсы системы.

d) Разрабатываются архитектурные представления и модели системы.

e) Концепции, свойства, характеристики, поведение, функции или ограничения, значимые для архитектурных решений системы, распределяются по архитектурным сущностям.

f) Определяются элементы системы и их интерфейсы.

g) Оцениваются кандидаты на архитектуру.

h) Достигается архитектурная основа для процессов на протяжении всего жизненного цикла.

i) Достигается согласование архитектуры с требованиями и проектными характеристиками.

j) Доступны любые вспомогательные системы или услуги, необходимые для определения архитектуры.

k) Разработана прослеживаемость элементов архитектуры к требованиям заинтересованных сторон и системы/программного обеспечения.

6.4.4.3 Деятельность и задачи

Проект должен реализовать следующие виды деятельности и задачи в соответствии с применимыми политиками и процедурами организации в отношении процесса определения архитектуры.

a) Подготовка к определению архитектуры. Деятельность состоит из следующих задач:

1) Изучение соответствующей информации и определение ключевых факторов архитектуры.

Примечания

1 Ключевые движущие силы определяются путем анализа: (a) исследований рынка, отраслевых прогнозов, планов продуктов конкурентов и научных выводов; (b) стратегии организации, концепции операций на уровне организации, политики и директив организации, нормативно-правовых ограничений и требований заинтересованных сторон; (c) концепции операций миссии или бизнеса, интересующей системы и соответствующей концепции функционирования системы, операционной среды, технологических дорожных карт и требований к системе/программному обеспечению; и (d) других факторов, влияющих на пригодность программной системы на протяжении ее жизненного цикла. Анализ ключевых факторов обычно основывается на анализе бизнеса или миссии, определении требований заинтересованных сторон и определении требований к системе/программному обеспечению.

2 Ключевые факторы архитектуры могут включать стили и модели архитектуры, элементы, принципы, такие как заменяемые компоненты, осуществимость реализации и интеграции; доступность компонентов COTS и открытых ресурсов; ресурсы данных для систем с интенсивным использованием данных; и последствия производительности. Эффект от выбора различных элементов дизайна может быть уменьшен, если программная система правильно спроектирована.

2) Определение проблемы заинтересованных сторон.

Примечания

1 Заинтересованные стороны первоначально определяются в процессе определения потребностей и требований заинтересованных сторон. Дополнительные заинтересованные стороны обычно определяются в процессе определения архитектуры. Опасения заинтересованных сторон, связанные с архитектурой, включают опасения за целостность системы, того, что программная система будет скомпрометирована преднамеренно или непреднамеренно с помощью фактора угрозы или станет причиной несчастных случаев в качестве источника повышенной опасности. Заинтересованные стороны. ожидания или ограничения часто связаны с этапами жизненного цикла системы, такими как использование (например, доступность, безопасность, эффективность, практичность, совместимость с существующими системами, доступность или риски для данных в системе), поддержка (например, возможность поддержки системы в течение прогнозируемого срока службы, управление устареванием), эволюция программной системы и ее окружения (например, адаптивность, масштабируемость, живучесть), производство (например, распространение, тестируемость) и выход из эксплуатации (например, удаление или сохранение конфиденциальных данных).

2 Проблемы, влияющие на архитектуру программных систем, включают ресурсы данных и последствия производительности для систем с интенсивным использованием данных, а также ограничения на использование аутсорсинговых, существующих, вновь разработанных, запатентованных, коммерчески доступных или открытых элементов программного обеспечения, включая лицензирование программного обеспечения. Хотя архитектура программного обеспечения в идеале не зависит от дизайна, возможность реализации архитектуры в доступной программной системе является существенным ограничением для большинства систем.

3) Определение дорожной карты, подхода и стратегии определения архитектуры.

Примечание – Включает определение возможностей для общения с заинтересованными сторонами, определение мероприятий по рассмотрению архитектуры, подход и критерии оценки, подход к измерению и методы измерения (процесс «Измерение»). Дорожная карта показывает, как архитектура будет развиваться до предполагаемого конечного состояния, и часто имеет более длительные сроки, чем для текущей интересующей системы. Подход – способ выполнения работы, например, как взаимодействовать с заинтересованными сторонами, как проверять результаты или где выполнять работу. Стратегия – систематический план действий по реализации подхода в соответствии с дорожной картой.

4) Определение критерия оценки архитектуры на основе проблем заинтересованных сторон и ключевых требований.

5) Определение и планирование необходимых вспомогательных систем или услуг для поддержки процесса определения архитектуры.

Примечание – Включает определение требований и интерфейсов для вспомогательных систем и услуг. Вспомогательные системы для определения архитектуры могут включать инструменты для совместной работы и разработки архитектуры, а также репозитории повторного использования архитектуры для таких аспектов, как архитектурные шаблоны, модели и эталонные архитектуры.

6) Получение или приобритение доступа к используемым системам или услугам.

Примечание – Процесс валидации используется для объективного подтверждения того, что вспомогательная система достигает целевого использования для своих вспомогательных функций. Процесс управления инфраструктурой поддерживает повторное использование вспомогательных систем.

b) Разработка точек зрения на архитектуру. Деятельность состоит из следующих задач:

1) Выбор, адаптация или разработка точек зрения и видов моделей с учетом интересов заинтересованных сторон.

2) Создание или определение потенциальных архитектурных структур, которые будут использоваться при разработке моделей и представлений.

Примечание – Некоторые архитектурные структуры определяют заинтересованные стороны и их проблемы, а также соответствующие точки зрения, которые учитывают эти проблемы, в то время как другие архитектурные структуры имеют более общее руководство. Точки зрения определяют виды моделей, которые должны быть использованы, и то, как результирующие модели могут быть использованы для создания представлений архитектуры. Более подробная информация о структуре архитектуры и практике описания архитектуры приведена в [ISO/IEC/IEEE 42010].

3) Фиксация обоснования выбора структуры (структур), точек зрения и видов моделей.

4) Выбор или разработка вспомогательных методов и инструментов моделирования.

Примечание – И [SWEBOK], и [ISO/IEC TR 24748-3] описывают методы моделирования, которые поддерживают определение архитектуры и определение дизайна элементов программного обеспечения.

c) Разработка моделей и представлений возможных архитектур. Деятельность состоит из следующих задач:

1) Определение контекста и границ программной системы с точки зрения интерфейсов и взаимодействий с внешними объектами.

Примечание – Задача в основном основывается на результатах процесса анализа бизнеса или миссии и выполняется одновременно с процессом определения потребностей и требований заинтересованных сторон. Она заключается в определении внешних по отношению к программной системе объектов (т.е. существующих и проектируемых систем, продуктов и услуг, составляющих контекст системы) и определении границ программной системы (т.е. взаимодействия с этими внешними объектами через интерфейсы, пересекающие границы). Внешние сущности включают необходимые вспомогательные системы. Процесс определения архитектуры определяет интерфейсы в той степени, которая необходима для поддержки основных архитектурных решений и понимания. Затем эти определения интерфейсов уточняются в процессе определения дизайна.

2) Определение архитектурных модулей и отношений между модулями, которые отвечают ключевым проблемам заинтересованных сторон и критическим требованиям к программной системе.

Примечание – Архитектура не обязательно учитывает все требования, а скорее только те требования к системе/программному обеспечению, которые определяют архитектуру. С другой стороны, в процессе определения дизайна рассматриваются и учитываются все требования. Иногда в процессе определения архитектуры возникают требования, которые считаются неуместными, недоступными или неприемлемыми. Это вопросы требований, которые решаются с помощью итераций процесса определения требований к системе/программному обеспечению. Также важно, чтобы архитектура учитывала основные проблемы заинтересованных сторон, поскольку не все из них будут отражены в требованиях.

3) Распределение понятия, свойств, характеристик, поведений, функций или ограничений, значимых для архитектурных решений программной системы, между архитектурными модулями.

Примечание – Выделяемые объекты могут быть физическими, логическими или концептуальными.

4) Выбор, адаптация или разработка модели вариантных архитектур программной системы.

Примечание – Обычно при определении архитектуры используются модели, которые наилучшим образом отвечают ключевым проблемам заинтересованных сторон. О том, как это можно сделать, см. в [ISO/IEC/IEEE 42010]. При определении архитектуры принято использовать логические и физические модели. Информация о логических и других моделях представлена в Приложении F.

5) Составление представления из моделей в соответствии с определенными точками зрения, чтобы выразить, как архитектура решает проблемы заинтересованных сторон и отвечает требованиям заинтересованных сторон и системы/программного обеспечения.

6) Согласование архитектурных моделей и взаимных концепций.

Примечание – Правила соответствия из фреймворков – один из способов установления гармонии между представлениями [ISO/IEC/IEEE 42010].

d) Соотношение архитектуры и проектного решения. Деятельность состоит из следующих задач:

1) Определение элементов программной системы, которые относятся к архитектурным объектам, и характер этих отношений.

Примечание – Иногда элементы программной системы изначально являются условными, пока не произойдет определение дизайна, поскольку это зависит от фактического дизайна (проектов), который должен быть выполнен. Иногда создается «эталонная архитектура» с использованием условных элементов системы как средство передачи архитектурного замысла и проверки реализуемости проекта.

2) Определение интерфейсов и взаимодействий между элементами программной системы и внешними модулями.

Примечание – Определяется на уровне детализации, необходимом для передачи архитектурного замысла, и может быть уточнено в процессе определения дизайна.

3) Разделение, согласование и распределение требований по архитектурным модулям и элементам системы.

4) Сопоставление элементов программной системы и архитектурных модулей с характеристиками проекта.

5) Определение принципов проектирования и эволюции программных систем.

***Пример – Принципы могут включать функциональную совместимость, использование выбранных моделей проектирования, простоту замены и модернизации элементов системы или уровни безопасности.***

e) Оценка вариантов архитектуры. Деятельность состоит из следующих задач:

1) Оценка каждого варианта архитектуры на соответствие ограничениям и требованиям.

2) Оценка каждого варианта архитектуры с точки зрения проблем заинтересованных сторон, используя критерии оценки.

Примечание – Процесс системного анализа и процесс управления рисками могут быть использованы для поддержки этой задачи.

3) Выбор предпочтительной архитектуры и фиксирование ключевых решений и обоснований.

Примечание – Для поддержки этой задачи можно использовать процесс управления принятием решений.

4) Определение базовой линии выбранной архитектуры.

Примечание – Базовая линия архитектуры состоит из моделей, представлений и других соответствующих описаний архитектуры.

f) Управление выбранной архитектурой. Деятельность состоит из следующих задач:

1) Формализация подхода к управлению архитектурой и определение связанных с управлением ролей и обязанностей, подотчетности и полномочий, относящиеся к проектированию, качеству, надежности и безопасности.

2) Согласование архитектуры с заинтересованными сторонами.

Примечание – Процесс валидации используется для подтверждения того, что архитектурные модели и представления отражают требования заинтересованных сторон, что проблемы заинтересованных сторон учтены, и для обеспечения того, чтобы будущие итерации архитектуры программной системы лучше учитывали проблемы заинтересованных сторон.

3) Поддержка согласованности и полноты архитектурных объектов и их архитектурных характеристик.

Примечание – Проверяемые модули являются не только техническими. Но также, например, юридические, экономические, организационные и операционные модули, которые обычно являются частью требований и проблем заинтересованных сторон.

4) Организация, оценивание и конролирование эволюции архитектурных моделей и представлений для обеспечения выполнения архитектурного замысла и правильной реализации архитектурного видения и ключевых концепций.

5) Поддержка стратегии определения и оценки архитектуры.

Примечание – Включает обновление архитектуры на основе технологического развития (например, устаревание), внедрения или опыта эксплуатации. Включает в себя управление внешними и внутренними интерфейсами, которые определены на этом уровне декомпозиции программной системы.

6) Поддержка прослеживаемости архитектуры.

Примечание – На протяжении всего жизненного цикла часто поддерживается прослеживаемость между архитектурными объектами или элементами (моделями, представлениями и точками зрения), требованиями (включая выделенные, декомпозированные и производные) и проблемами заинтересованных сторон, проектированием программной системы, определением интерфейсов, результатами анализа и методами или способами проверки.

7) Предоставление ключевых аспектов и информационных элементов, которые были выбраны для базовых линий.

Примечание – Процесс управления конфигурацией используется для создания и поддержания элементов конфигурации и базовых линий. Процесс определяет варианты для базовой линии. Процесс управления информацией контролирует информационные элементы, такие как описания архитектуры (модели архитектуры, представления архитектуры, оценки и прослеживаемость).

6.4.5 Процесс определения проектного решения

6.4.5.1 Цель

Целью процесса определения проектного решения является предоставление достаточно подробных данных и информации о системе и ее элементах для обеспечения возможности реализации в соответствии с архитектурными модулями, определенными в моделях и представлениях архитектуры системы.

Для программных систем деятельность по проектированию обычно итерируется с деятельностью по определению требований к системе/программному обеспечению и определению архитектуры. Определение дизайна применяется итеративно и постепенно для разработки детального дизайна, включая элементы программного обеспечения, интерфейсы, базы данных и пользовательскую документацию. Проектирование программного обеспечения осуществляется одновременно с реализацией, интеграцией, верификацией и валидацией программного обеспечения. В приложении H рассматривается проектирование программного обеспечения с использованием гибких методов. В ходе проектирования и реализации дальнейшее применение процесса уточняет распределение изменяющихся требований между элементами программного обеспечения.

Примечания

1 Процесс определения проектного решения определяется требованиями, которые были проверены с помощью архитектуры и более детального анализа осуществимости. Архитектура фокусируется на пригодности, жизнеспособности и желательности, в то время как дизайн фокусируется на совместимости с технологиями и другими элементами проектного решения, а также на возможности реализации и интеграции. Эффективная архитектура является максимально независимой от проектного решения, чтобы обеспечить максимальную гибкость в торговом пространстве проектного решения.

2 Процесс обеспечивает обратную связь с архитектурой программной системы для закрепления или подтверждения распределения, разделения и выравнивания архитектурных объектов.

6.4.5.2 Результаты

В результате успешной реализации процесса определения дизайна:

a) Определяются проектные характеристики каждого элемента системы.

b) Распределяются требования к системе/программному обеспечению по элементам системы.

c) Выбираются или определяются средства обеспечения проектирования, необходимые для определения конструкции.

d) Определяются или уточняются интерфейсы между элементами системы, составляющими систему.

e) Оцениваются альтернативы проектирования для элементов системы.

f) Разрабатываются артефакты проектирования.

g) Доступны любые вспомогательные системы или услуги, необходимые для определения проектного решения.

h) Устанавливается прослеживаемость проектных характеристик до архитектурных элементов архитектуры системы.

Примечание – Определение проектного решения учитывает применимые технологии и их вклад в системное решение. Проектирование обеспечивает уровень определения «от реализации до реализации», такой как чертежи, диаграммы состояний, истории и подробное описание проекта. Для элементов программного обеспечения этот процесс может привести к подробному описанию дизайна, который может быть проверен на соответствие требованиям и архитектуре программного обеспечения. Даже если проект программного обеспечения не полностью определен в формальном описании, он достаточно подробен, чтобы позволить реализацию (построение) программного обеспечения и планирование.

6.4.5.3 Деятельность и задачи

Проект должен выполнить следующие мероприятия и задачи в соответствии с применимыми политиками и процедурами организации в отношении процесса.

Примечание – [SWEBOK] содержит подробное обсуждение проектирования программного обеспечения. В этой области знаний рассматриваются основы, ключевые вопросы, стратегии и методы проектирования, а также обозначения для проектирования.

a) Подготовка к определению проектного решения программной системы. Деятельность состоит из следующих задач:

1) Определение стратегии описания проектного решения, соответствующего выбранной модели жизненного цикла и предполагаемым артефактам дизайна.

Примечание – Стратегия проектирования программного обеспечения может включать первоначальную или постепенную декомпозицию на элементы системы; создание различных представлений автоматизированных процедур, структур данных и систем управления; выбор шаблонов проектирования или постепенное более детальное определение объектов и их взаимосвязей.

2) Выбор и приоритизациия принципов и характеристик проектного решения.

Примечание – Принципы проектирования включают в себя такие идеи управления, как абстракция, модуляция и инкапсуляция, разделение интерфейса и реализации, параллелизм и сохранение данных. Соображения безопасности включают принцип наименьших привилегий, многоуровневую защиту, ограниченный доступ к системным сервисам и другие соображения для минимизации и защиты поверхности атаки системы. Характеристики проектирования включают, например, доступность, отказоустойчивость и стойкость, масштабируемость, удобство использования, емкость и производительность, тестируемость, переносимость и доступность.

3) Определение и планирование необходимых вспомогательных систем или услуг обеспечивающих поддержку определения проектного решения.

Примечание – Включает определение требований и интерфейсов для вспомогательных систем. Вспомогательные системы для определения дизайна включают выбор программных и системных платформ, языков программирования, обозначений и инструментов для совместной работы и разработки дизайна, хранилищ повторного использования дизайна (для линий продуктов, шаблонов проектного решения и аспектов проектного решения) и стандартов проектирования.

4) Получение или приобретение доступа к используемым системам или услугам, обеспечивающим возможность их использования.

Примечание – Процесс валидации используется для объективного подтверждения того, что вспомогательная система достигает целевого использования для своих вспомогательных функций.

b) Создание проектов, относящихся к каждому элементу программной системы. Деятельность состоит из следующих задач:

1) Преобразование архитектурных и проектных характеристик в проектирование элементов программной системы.

Примечание – Характеристики относятся к физическим и логическим элементам системы, таким как структуры баз данных, резервы памяти и хранения, программные процессы и средства управления, внешние интерфейсы, такие как пользовательские интерфейсы, или услуги. [ISO 9241- 210] предоставляет руководящие принципы проектирования, ориентированного на человека/эргономичного проектирования.

2) Определение и подготовка или получение необходимых средств обеспечения проектирования.

Примечание – Средства проектирования включают модели, уравнения, алгоритмы, вычисления, формальные выражения и значения параметров, шаблоны и эвристики, которые связаны с характеристиками проектирования с использованием адекватного представления, такого как рисунки, логические диаграммы, блок-схемы, кодовые соглашения, логические схемы, информационные модели, бизнес-правила, профили пользователей, сценарии, случаи использования или истории пользователя, а также таблицы метрик и их значений, например, функциональные точки или точки истории пользователя.

3) Изучение альтернативных вариантов проектных решений и возможность их реализации.

Примечания

1 Для системы программного обеспечения и элементов программного обеспечения, как правило, исследуются повторное использование, адаптация, аутсорсинговое обслуживание или новая разработка.

2 Оценка реализации проектных характеристик. Если оправдано результатами оценки, возможны другие альтернативные варианты проектного решения или компромисс в архитектуре или требованиях, если характеристики проектного решения нецелесообразно реализовать.

4) Уточнение или определение интерфейсов между элементами программной системы и с внешними модулями.

Примечание – Интерфейсы идентифицируются и определяются в процессе определения архитектуры (6.4.4) до уровня или степени, необходимой для замысла и понимания архитектуры. Они уточняются в процессе определения дизайна на основе характеристик дизайна, интерфейсов и взаимодействий элементов программного обеспечения с другими элементами, составляющими программную систему, и с внешними сущностями. Иногда выявляются и определяются дополнительные интерфейсы, которые не были рассмотрены в определении архитектуры.

5) Создание аспектов проектирования.

Примечание – Задача формализует проектные характеристики элементов программной системы с помощью выделенных аспектов в зависимости от технологии реализации. Примеры аспектов включают прототипы, модели данных, псевдокод, диаграммы отношений сущностей, варианты использования, матрицы ролей и привилегий пользователей, спецификации интерфейсов, описания сервисов и процедуры. Артефакты дизайна разрабатываются, приобретаются или модифицируются для выбранных альтернатив. Данные связаны с подробными допустимыми запасами для реализации (если это актуально для этого процесса или итерации задачи).

c) Оценка альтернатив для получения элементов программной системы. Деятельность состоит из следующих задач:

1) Определение технологий, необходимых для каждого элемента, составляющего программную систему.

Примечание – Для отдельно взятого элемента программной системы иногда используется несколько технологий, например, наличие интернета, встроенные системы, адаптация программного обеспечения с открытым исходным кодом, роли человека-оператора.

2) Определение альтернативных вариантов для элементов программной системы.

Примечание – Альтернативы включают новые разработанные и созданные изделия; адаптацию существующих линий продукции, компонентов, объектов или услуг; или приобретение или повторное использование неразработанных изделий (НРИ). НРИ включают пакеты или элементы COTS (Коммерческая готовая продукция) или FOSS (Бесплатное программное обеспечение с открытым исходным кодом), повторное использование предыдущей конструкции или существующих активов, включая предметы, предоставленные покупателем.

3) Оценка каждого альтернативного варианта по критериям, разработанным на основе ожидаемых характеристик проектного решения и требований к элементам, чтобы определить пригодность для предполагаемого применения.

Примечание – Решение «сделать или купить» и вытекающий из него подход к реализации и интеграции обычно подразумевают компромисс между критериями проектирования, включая стоимость. При выборе проектного решения учитываются вспомогательные системы, необходимые для тестирования альтернативного кандидата (проектирование и разработка на основе тестирования), и устойчивость в течение срока службы системы, включая затраты на обслуживание. Процесс технического обслуживания может быть использован для определения пригодности конструкции для долгосрочного технического обслуживания и устойчивости.

4) Выбор предпочтительных альтернатив среди возможных проектных решений для элементов программной системы.

Примечание – Процесс системного анализа можно использовать для анализа и оценки, чтобы поддержать процесс управления решениями при выполнении выбора. Анализ дизайна проводится с использованием процесса валидации.

d) Управление проектированием. Эта деятельность состоит из следующих задач:

1) Фиксация проектного решения и обоснование.

Примечание – Собираемая информация включает элементы программной системы и связанные с ними требования и проектные данные, например, для программных элементов, внутренних и внешних интерфейсов, структур данных, требований к реализации и тестированию, данных об объединении единиц для интеграции и тестовых случаев. Обоснование включает в себя информацию об основных вариантах реализации и вспомогательных средствах. Результирующий проект контролируется в соответствии со стратегией.

2) Установка прослеживаемости между элементами детального проектирования, требованиями к системе/программному обеспечению и архитектурными элементами архитектуры программной системы.

Примечания

1 Задача облегчает обеспечение обратной связи с процессом определения архитектуры для возможных изменений, например, для изменения распределения элементов программной системы с целью получения ожидаемых архитектурных характеристик; или, возможно, для изменения ожидаемых архитектурных характеристик из-за факторов, обнаруженных в процессе проектирования, или для того, чтобы заинтересованные стороны знали о потенциальных последствиях.

2 На протяжении всего жизненного цикла поддерживается двунаправленная прослеживаемость между проектом и методами или методиками верификации, а также требованиями к элементам программной системы. Распределение и свойства проектирования назначаются элементам программного обеспечения, программным единицам и связанным с ними аспектами на достаточно подробном уровне, чтобы позволить тестирование и реализацию программного обеспечения, включая строительство.

3) Определение статуса проектирования программной системы и элементов.

Примечания

1 Процесс оценки используется для установления показателей полноты и качества разработки по мере ее выполнения. Процессы верификации и валидации используются для проверки и валидации детального проекта и реализации

2 Включает периодическую оценку проектных характеристик в случае эволюции программной системы и ее архитектуры, а также прогнозирование потенциального устаревания компонентов и технологий, их замены другими с течением времени в жизненном цикле программной системы и последствий для проектного определения. Процесс управления рисками обычно применяется для оценки рисков в стратегии проектирования, первоначальном проекте и развивающемся проекте.

4) Предоставление ключевых артефактов и информационных элементов, которые были выбраны для базовых линий.

Примечание – Процесс управления конфигурацией используется для создания и поддержания элементов конфигурации и базовых линий для таких аспектов, как модели проектирования. Процесс определяет кандидатов для базовой линии, а процесс управления информацией контролирует информационные элементы, такие как описания дизайна и спецификации.

6.4.6 Процесс системного анализа

6.4.6.1 Цель

Цель процесса системного анализа - обеспечить строгую основу данных и информации для технического понимания, чтобы помочь в принятии решений на протяжении всего жизненного цикла.

Процесс системного анализа применяется к разработке исходных данных, необходимых для любой технической оценки. Он может обеспечить уверенность в полезности и целостности системных требований, архитектуры и дизайна. Системный анализ охватывает широкий спектр различных аналитических функций, уровней сложности и степени строгости. Он включает математический анализ, моделирование, имитацию, эксперименты и другие методы для анализа технических характеристик, поведения системы, осуществимости, доступности, критических характеристик качества, технических рисков, стоимости жизненного цикла, а также для проведения анализа чувствительности потенциального диапазона значений параметров на всех этапах жизненного цикла. Он используется для широкого спектра аналитических потребностей, касающихся операционных концепций, определения значений требований, разрешения конфликтов требований, оценки альтернативных архитектур или элементов системы, а также оценки инженерных стратегий (интеграция, верификация, валидация и техническое обслуживание). Формальность и строгость анализа зависит от критичности информационной потребности или поддерживаемого рабочего продукта, объема доступной информации/данных, размера проекта и графика получения результатов.

Примечание – Процесс системного анализа может быть использован для всей программной системы или любого элемента. Процесс часто используется в сочетании с процессом управления принятием решений.

6.4.6.2 Результаты

В результате успешного внедрения процесса системного анализа:

a) Определены необходимые системные анализы.

b) Утверждаются предположения и результаты системного анализа.

c) Результаты системного анализа предоставляются для принятия решений.

d) Доступны любые вспомогательные системы или услуги, необходимые для системного анализа.

e) Устанавливается прослеживаемость результатов системного анализа.

6.4.6.3 Деятельность и задачи

Проект должен реализовать следующие виды деятельности и задачи в соответствии с применимыми политиками и процедурами организации в отношении процесса системного анализа.

a) Определение стратегии системного анализа и подготовка к системному анализу. Эта деятельность состоит из следующих задач:

1) Определение проблемы или вопроса, требующих анализа.

Примечание – Сюда входят технические, функциональные и нефункциональные цели анализа. К нефункциональным целям относятся критические характеристики качества, различные свойства, зрелость технологии и технические риски. Постановка проблемы или вопроса, на который должен ответить анализ, имеет важное значение для определения целей анализа, а также ожиданий и полезности результатов.

2) Определение заинтересованных сторон анализа.

3) Определение объема, цели и уровня достоверности анализа.

Примечание – Необходимый уровень достоверности (точности или аккуратности) является фактором для определения соответствующего уровня строгости.

4) Выбор методов поддержки анализа.

Примечание – Методы выбираются с учетом времени, стоимости, точности, технических факторов и критичности анализа. Методы анализа имеют широкий диапазон уровней строгости и включают экспертную оценку, расчеты на рабочем листе, параметрические оценки и расчеты, исторические данные и анализ тенденций, инженерные модели, моделирование, визуализацию и создание прототипов. Из-за ограничений по стоимости и срокам в большинстве проектов системный анализ проводится только для критических характеристик.

5) Определение и планирование необходимых вспомогательных систем или услуг, необходимых для поддержки анализа.

Примечание – Задача включает определение требований и интерфейсов для вспомогательных систем. К вспомогательным системам системного анализа относятся инструменты, соответствующие модели и потенциальные хранилища данных, необходимые для поддержки анализа. Выбранные методы будут основным фактором в определении того, какие инструменты подходят для поддержки анализа. Также включает определение доступности многократно используемых или других соответствующих моделей и данных, или ресурсов.

6) Получение или приобретение доступа к используемым системам или услугам.

Примечание – Процесс «Управление инфраструктурой» позволяет предоставлять услуги системного анализа. Процесс используется для объективного подтверждения того, что вспомогательная система достигает целевого использования для своих вспомогательных функций.

7) Сбор данных и исходных материалов, необходимых для анализа.

b) Выполнение системного анализа. Эта деятельность состоит из следующих задач:

1) Определение и проверка контекстов и допущений.

2) Применение выбранных методов анализа для проведения необходимого анализа.

3) Проверка результатов анализа на качество и обоснованность.

Примечание – Результаты согласовываются с соответствующими анализами, которые были выполнены ранее.

4) Установка выводов и рекомендаций.

Примечание – Для выполнения этой задачи определяются и привлекаются соответствующие профильные эксперты и заинтересованные стороны.

5) Фиксация результатов системного анализа,

c) Управление системным анализом. Эта деятельность состоит из следующих задач:

1) Поддержка прослеживаемости результатов анализа.

Примечание – На протяжении всего жизненного цикла поддерживается двунаправленная прослеживаемость между результатами анализа и любым элементом программной системы, для которого анализ поддерживает решение или дает обоснование (например, значения требований к системе/программному обеспечению, альтернативные варианты архитектуры). Этому часто способствует соответствующее хранилище данных.

2) Предоставление ключевых аспектов и информационных элементов, которые были выбраны для базовых линий.

Примечание – Процесс управления конфигурацией используется для создания и поддержания элементов конфигурации и базовых линий. Процесс определяет кандидатов для базовой линии, а процесс управления информацией управляет информационными элементами. Для этого процесса результаты анализа или отчеты являются типичными информационными элементами, которыми управляют.

6.4.7 Процесс внедрения

6.4.7.1 Цель

Целью процесса реализации является реализация заданного элемента системы.

Процесс преобразует требования, архитектуру и дизайн, включая интерфейсы, действия, которые создают элемент системы в соответствии с практикой выбранной технологии реализации с использованием соответствующих технических специальностей или дисциплин. Результатом этого процесса является элемент системы, который удовлетворяет заданным системным требованиям (включая выделенные и производные требования), архитектуре и дизайну.

Для программных систем целью процесса реализации является реализация элемента программной системы.

Элементы программной системы могут включать аппаратные средства, программное обеспечение и услуги. Для реализации программного обеспечения этот процесс преобразует заданные конструкции, поведение, интерфейсы и ограничения реализации в действия, которые создают элемент программной системы, реализованный в виде программного продукта или услуги, также известный как "элемент программного обеспечения". Результатом реализации программного обеспечения является элемент программного обеспечения, который удовлетворяет заданным требованиям путем проверки и требованиям заинтересованных сторон путем валидации. Реализация программного обеспечения включает различные комбинации конструирования (кодирование вновь созданных элементов программного обеспечения), приобретение новых пакетов программного обеспечения (например, из открытых источников, коммерческих или организационных источников) или повторное использование существующих элементов (с модификацией или без нее).

Реализация программного обеспечения включает использование процессов Соглашения для получения неразрабатываемых предметов (НРИ), таких как аппаратное обеспечение и операционные системы (платформа) или вспомогательные системы и услуги. Внедрение программного обеспечения обычно выполняется одновременно с интеграцией программного обеспечения. Внедрение обычно выполняется вместе со всеми процессами технического управления и многими техническими процессами, в частности:

a) Процесс верификации, который обеспечивает объективное свидетельство того, что реализация программного обеспечения соответствует установленным требованиям и выявляет аномалии (ошибки, дефекты, неисправности) в связанных с реализацией информационных элементах (например, требования к системе/программному обеспечению, архитектура, дизайн или другие описания), процессах, элементах программного обеспечения, элементах, блоках;

b) Процесс валидации, который подтверждает, что реализация соответствует требованиям для конкретного предполагаемого использования рабочего продукта программного обеспечения.

6.4.7.2 Результаты

В результате успешной реализации процесса внедрения:

a) Определяются ограничения реализации, влияющие на требования, архитектуру или дизайн.

b) Элемент системы реализуется.

c) Элемент системы упаковывается или хранится.

d) Доступны любые вспомогательные системы или услуги, необходимые для реализации.

e) Устанавливается прослеживаемость.

6.4.7.3 Деятельность и задачи

В рамках проекта должны быть выполнены следующие мероприятия и задачи в соответствии с применимыми политиками и процедурами организации в отношении процесса реализации.

a) Подготовка к внедрению. Эта деятельность состоит из следующих задач:

1) Определение стратегии внедрения, учитывая следующее:

i) политики и стандарты разработки, включая стандарты, регулирующие применимые методы обеспечения безопасности, защиты, конфиденциальности и охраны окружающей среды; стандарты программирования или кодирования; правила модульного тестирования; и специфические для конкретного языка стандарты реализации функций безопасности;

ii) методы определения уровня, источника и пригодности повторно используемых элементов системы и безопасности цепи поставок для повторно используемого или адаптированного программного обеспечения;

iii) процедуры и методы разработки (конструирования) ПО и разработки модульных тестов; использование экспертных оценок, модульных тестов и прохождений во время внедрения;

iv) использование контроля CM во время разработки программного обеспечения;

v) предложения по управлению изменениями для ручных процессов;

vi) приоритеты внедрения для поддержки миграции и перехода данных и программного обеспечения, а также вывода из эксплуатации устаревших систем;

vii) создание ручных или автоматизированных процедур тестирования для проверки того, что программная единица соответствует требованиям до создания программной единицы (разработка на основе тестирования);

viii) комплексные или специализированные среды разработки и поддержки жизненного цикла для реализации и управления требованиями, моделями и прототипами, поставляемыми элементами системы или программного обеспечения, а также тестовыми спецификациями и тестовыми случаями.

Примечание – Стратегия реализации записывается в SDP или SEMP проекта, а иногда и в PMP.

2) Определите ограничения со стороны стратегии реализации и технологии реализации на требования к системе/программному обеспечению, характеристики архитектуры, характеристики проектирования или методы реализации.

Примечания

1 Ограничения включают текущие или ожидаемые ограничения выбранной технологии внедрения (например, для программного обеспечения, операционной системы, системы управления базами данных, веб-сервисов), материалы или элементы системы, предоставленные покупателем для адаптации, и ограничения, возникающие в результате использования необходимых систем, способствующих внедрению.

2 Стратегия реализации программного обеспечения определяет и распределяет критерии «реализации до», например, характеристики архитектуры и дизайна программного обеспечения, требования к системе/программному обеспечению, включая обеспечение безопасности программного обеспечения, соображения удобства использования, управление конфигурацией, прослеживаемость или другие условия, которые должны быть удовлетворены. Критерии могут уточнить соответствующие уровни агрегации единиц, спецификации и ограничения.

3) Определение и планирование необходимых и отдельных программных сред, включая вспомогательные системы или услуги, необходимые для поддержки разработки и тестирования.

Примечание – Внедрение программного обеспечения использует отдельные среды, которые отделены под контролем конфигурации от операционной (производственной) среды. Общий процесс внедрения, вспомогательные системы и услуги включают комплексные или специализированные среды разработки и поддержки жизненного цикла для реализации и управления требованиями, моделями и прототипами, поставляемыми элементами, а также тестовыми средами, спецификациями и тестовыми случаями; симуляторы для внешних систем, обучающие системы; и системы управления контентом для пользовательской документации.

4) Получение или приобритение доступа к программным средам и другим вспомогательным системам или услугам.

Примечание – Процесс валидации используется для объективного подтверждения того, что система, обеспечивающая интеграцию, достигает целевого использования для своих функций, обеспечивающих интеграцию.

b) Выполнение внедрения. Деятельность состоит из следующих задач:

Примечание – На протяжении всего процесса внедрения процесс верификации используется для объективного подтверждения соответствия элементов системы требованиям. Процесс валидации используется для объективного подтверждения того, что элемент пригоден для использования в предполагаемой операционной среде в соответствии с требованиями заинтересованных сторон.

1) Реализация или адаптация элементов программного обеспечения в соответствии со стратегией, ограничениями и определенными процедурами реализации.

Примечания

1 Элементы программного обеспечения приобретаются, определяются для повторного использования из активов организации или разрабатываются (создаются). Приобретаемые элементы программного обеспечения могут варьироваться от простой покупки продукта в соответствии с правилами закупок организации или проекта до сложного приобретения программной системы, в котором задействованы процессы приобретения и поставки. Адаптация включает конфигурацию программных элементов, которые используются повторно или модифицируются. Создание может включать кодирование программного обеспечения, адаптивное повторное использование и интеграцию существующих модулей, рефакторинг, разработку базы данных и создание ручных или автоматизированных процедур тестирования для каждого модуля.

2 Для разрабатываемых элементов программного обеспечения на самом низком уровне реализации создаются исполняемые единицы программного обеспечения (часто с соответствующими структурами данных, интерфейсами прикладного программирования, описаниями услуг, пользовательской документацией, тестовыми случаями или другими элементами), контролируются, предоставляются уполномоченным ролям и хранятся в соответствии с процедурами CM для аспектов разработки.

3 В документе [SWEBOK] содержится подробное обсуждение построения программного обеспечения. В этой области знаний рассматриваются основы, управление, измерение, практические предложения (например, проектирование конструкции, языки, тестирование, повторное использование и интеграция), технологии конструирования (например, объектно-ориентированные, обработка ошибок и исключений, исполняемые модели, распределенное программное обеспечение), а также инструменты и среды.

2) Реализация или адаптация аппаратных элементов программных систем.

Примечание – Аппаратные элементы приобретаются или изготавливаются с использованием применимых методов, соответствующих выбранной технологии физической реализации и материалов. При необходимости аппаратные элементы проверяются на соответствие заданным системным требованиям и критическим характеристикам качества. В случае повторного внедрения элементов системы (например, массовое производство, замена элементов системы) процедуры внедрения и процессы изготовления определяются и могут быть автоматизированы для достижения последовательной и повторяемой производительности. Некоторые общие аппаратные элементы в программных системах включают интеграцию приобретенных систем COTS, специальные модификации, например, для испытательных или эксплуатационных сред, и аппаратные средства управления со встроенным программным обеспечением.

3) Реализация или адаптация сервисных элементов программных систем.

Примечание – Элементы услуг включают набор услуг, которые должны быть предоставлены. [ISO/IEC 20000 (IEEE Std 20000)] применяется к управлению элементами системы, реализованными в услугах, включая стратегию, проектирование и переход. При необходимости элементы услуг проверяются на соответствие системным требованиям и критериям услуг. Например, элементы оперативных ресурсов проверяются на соответствие системным требованиям и оперативной концепции. Элементы услуг могут включать сетевые коммуникации, обучение, услуги по упаковке и распространению программного обеспечения, услуги по настройке программного обеспечения под конкретные нужды заказчика, мониторинг операционной деятельности и безопасности, а также помощь пользователям.

4) Оценка программного блока и связанных с ним данных или другой информации в соответствии со стратегией реализации и критериями.

Примечание

1 Критерии оценки обычно включают удовлетворение требований к блоку и критериев тестирования, покрытие тестов блока, требования к прослеживаемости, согласованность с требованиями или дизайном элемента программного обеспечения, согласованность внутренних требований к блоку, и осуществимость для дальнейшей деятельности процесса, например, интеграции, верификации, валидации, эксплуатации и обслуживания.

2 Используйте Управление результатами деятельности по внедрению для регистрации структуры и устранения аномалий.

5) Группировка и хранение элемента программной системы.

Примечание – Элемент программной системы необходим для достижения непрерывности его характеристик. Передача и хранение, а также их продолжительность могут влиять на указанное сдерживание. Для программного обеспечения основная копия внедренного программного обеспечения (электронная или на физических носителях) хранится в контролируемом месте и предоставляется уполномоченным лицам (например, для использования в процессах интеграции и перехода). Информация о конфигурации и продукте фиксируется процессами управления конфигурацией и управления информацией, когда элемент хранится.

6) Запись объективных доказательств того, что элемент программной системы соответствует требованиям.

Примечание – Доказательства предоставляются в соответствии с договорами поставки, законодательством и политикой организации. Доказательства включают модификации элемента, сделанные в связи с изменениями в обработке или несоответствиями, обнаруженными в ходе процессов верификации и валидации. Объективные доказательства являются частью базовой конфигурации элемента, установленной в процессе управления конфигурацией, и включают результаты тестирования, анализа, инспекций, проходных мероприятий, демонстраций, обзоров продукции или технических обзоров или других мероприятий по проверке.

c) Управление результатами внедрения. Деятельность состоит из следующих задач:

1) Фиксация результатов внедрения и обнаруженные ошибки.

Примечание – Сюда входят ошибки, связанные со стратегией внедрения, системами, обеспечивающими внедрение, или неправильным определением программной системы. Процессы оценки и контроля проекта и обеспечения качества используются для анализа данных с целью выявления первопричины, принятия мер по исправлению или улучшению, а также для регистрации полученных результатов.

2) Поддержка прослеживаемости реализованных элементов программной системы.

Примечания

1 Для поддержки прослеживаемости на протяжении всего жизненного цикла при эксплуатации и обслуживании ведется учет ресурсов лицензий на программное обеспечение и других системных активов в цепочке поставок. Процессы управления информацией и управления конфигурацией используются для поддержания условий поддержки лицензий и технического обслуживания для программного приложения и его необходимой инфраструктуры (хост-системы). Стандарты [ISO/IEC 19770] содержат требования к системе управления ИТ-активами.

2 Поддерживается двунаправленная прослеживаемость между реализованными элементами и архитектурой программной системы; проектом и связанными с ним требованиями, включая требования к интерфейсу и определения, необходимые для реализации; а также планами, процедурами и результатами валидации и верификации.

3) Предоставление ключевых аспектов и информационных элементов, которые были выбраны для базовых линий.

Примечание – Процесс управления конфигурацией используется для создания и поддержки элементов конфигурации и базовых показателей. Этот процесс определяет кандидатов для базового уровня, а процесс управления информацией контролирует информационные единицы. Для этого процесса элементы системы программного обеспечения (например, исходный код), пакеты программного обеспечения и результаты модульного тестирования являются типичными артефактами, которые являются базовыми.

6.4.8 Процесс интеграции

6.4.8.1 Цель

Целью процесса интеграции является синтез набора элементов системы в реализованную систему (продукт или услугу), которая удовлетворяет требованиям к системе/программному обеспечению, архитектуре и дизайну.

В ходе этого процесса осуществляется сборка реализованных элементов системы. Определяются и активируются интерфейсы для обеспечения взаимодействия элементов системы в соответствии с их назначением. Процесс интегрирует вспомогательные системы с интересующей системой для облегчения взаимодействия.

Интеграция программных систем итеративно объединяет реализованные элементы программных систем для формирования полных или частичных системных конфигураций с целью создания продукта или услуги. Интеграция программного обеспечения выполняется ежедневно или непрерывно на этапах разработки и сопровождения с использованием автоматизированных инструментов. Непрерывная интеграция предполагает включение или замену и архивирование элементов в библиотеках программного обеспечения под контролем CM.

Примечание – Интерфейсы определяются процессами определения архитектуры и определения дизайна. Процесс интеграции координируется с этими другими процессами для проверки того, что определения интерфейсов, как реализованные и интегрированные, являются адекватными и что они учитывают потребности интеграции.

6.4.8.2 Результаты

В результате успешного внедрения процесса интеграции:

a) Определяются интеграционные ограничения, влияющие на системные требования, архитектуру или дизайн, включая интерфейсы.

b) Определены подходы и контрольные точки для правильной работы собранных интерфейсов и системных функций.

c) Доступны любые вспомогательные системы или услуги, необходимые для интеграции.

d) Система, состоящая из реализованных элементов системы, интегрирована.

e) Проверяются интерфейсы между реализованными элементами системы, составляющими систему.

f) Проверяются интерфейсы между системой и внешней средой.

g) Выявляются результаты интеграции и ошибок.

h) Устанавливается прослеживаемость интегрированных элементов системы.

6.4.8.3 Деятельность и задачи

В рамках проекта должны быть выполнены следующие мероприятия и задачи в соответствии с применимыми политиками и процедурами организации в отношении процесса интеграции.

a) Подготовка к интеграции. Деятельность состоит из следующих задач:

1) Определение стратегии интеграции.

Примечания

1 Интеграция создает последовательности все более полных элементов системы программного обеспечения или конфигураций элементов программного обеспечения. Она зависит от доступности соответствующих элементов программной системы и соответствует стратегии изоляции и диагностики неисправностей. Последовательное применение процесса интеграции, процесса верификации и, при необходимости, процесса валидации повторяется для элементов в структуре системы до тех пор, пока не будет реализована интересующая система. Симуляторы или прототипы используются для элементов системы, которые еще не реализованы, например, для получения данных от взаимодействующих систем. Интеграция реализованных элементов программной системы основывается на приоритетах соответствующих требований и определении архитектуры, фокусируясь на интерфейсах, при этом минимизируя время интеграции, стоимость и риски. Интеграция программных систем поддерживает контроль версий с помощью процесса управления конфигурацией для выбора элементов конфигурации, подлежащих интеграции.

2 Для интеграции программного обеспечения стратегия интеграции согласуется со стратегией регрессии, которая применяется для повторной проверки элементов программного обеспечения при изменении связанных единиц программного обеспечения (и потенциально связанных с ними требований, дизайна и пользовательской документации).

3 Определение стратегии интеграции единиц и элементов программного обеспечения сопровождается определением стратегии для других процессов, которые происходят одновременно, таких как:

i) Процесс внедрения, помогающий обеспечить своевременную координацию задач процесса внедрения и интеграции и обеспечивающих систем, например, комбинированных сред разработки и тестирования программного обеспечения для поддержки автоматизированного или непрерывного внедрения и интеграции программных модулей и элементов.

ii) Процесс верификации для обеспечения объективных доказательств того, что интегрированное программное обеспечение соответствует заданным требованиям и выявления аномалий (ошибок, дефектов, неисправностей) в связанных с интеграцией информационных элементах (например, требованиях к системе/программному обеспечению, архитектуре, проектированию, тестированию или других описаниях), процессах, элементах программного обеспечения, элементах, единицах.

iii) Процесс валидации для подтверждения того, что рабочий продукт соответствует требованиям для конкретного предполагаемого использования интегрированной функции программного обеспечения..

iv) Процесс обеспечения качества для поддержки процессов интеграции и аудитов и инспекций рабочих продуктов, а также для решения проблем, несоответствий или отчетности и обработки инцидентов.

4 Стратегия интеграции фиксируется в плане, например, в интеграционном плане или в SDP или SEMP проекта.

2) Выявление и определение критериев интеграции и точек, в которых будет проверяться правильность работы и целостность интерфейсов и выбранных функций программной системы

Примечания

1 Детальная проверка интерфейсов выполняется с помощью процесса верификации. Интеграция программного обеспечения включает объединение элементов программного обеспечения, в результате чего получается набор интегрированных элементов программного обеспечения, который соответствует проекту программного обеспечения и удовлетворяет функциональным и нефункциональным требованиям к системе/программному обеспечению в эквиваленте операционной среды.

2 Для проектов с участием нескольких поставщиков или групп разработчиков, доступность элементов программной системы для интеграции обычно является частью графика проекта с этапами в рамках процесса оценки и контроля проекта. Интеграция происходит по мере проверки функциональности, производительности и пригодности программного обеспечения для использования в конкретных условиях или на конкретной платформе. На основных этапах интеграции, например, при завершении этапа, элемента или версии, обычно проводятся контрольные точки для обзора и проверки с заинтересованными сторонами. Частота этих обзоров связана с выбранной моделью жизненного цикла и методом разработки.

3) Определение и планирование необходимых вспомогательных систем или услуг для поддержки интеграции.

Примечание – Включает определение требований и интерфейсов для вспомогательных систем. К вспомогательным системам для интеграции относятся средства интеграции, специализированное оборудование, системы обучения, системы отчетности о несоответствиях, тренажеры, измерительные приборы и безопасность окружающей среды. Для программного обеспечения это может включать комплекты регрессионных тестов и систем CM для интегрированного тестирования программных систем, системы отчетности об инцидентах и проблемах, симуляторы, представляющие внешние системы или неразработанные элементы, и системы управления библиотекой программного обеспечения для операций разработки. Необходимо выявить и определить изменения или специализации, необходимые для вспомогательных систем для поддержки задач интеграции. Как правило, вспомогательные системы или услуги, используемые для интеграции на этапах разработки, могут также помочь в поддержке интеграции элементов системы по мере того, как программная система и вспомогательные среды переходят в рабочее состояние. Такой подход "DevOps" поддерживает итерационные процессы внедрения, интеграции, верификации, перехода, валидации, эксплуатации и обслуживания программной системы.

4) Получение или приобретение доступа к вспомогательным системам или услугам, которые будут использоваться для поддержки интеграции.

Примечание – Процесс валидации используется для объективного подтверждения того, что система, обеспечивающая интеграцию, достигает запланированного использования для своих функций, обеспечивающих интеграцию.

5) Определение ограничений для интеграции, которые должны быть включены в требования, архитектуру или дизайн системы/программного обеспечения.

Примечание – Сюда входят такие требования, как доступность, безопасность цепи поставок, безопасность для интеграторов, необходимые взаимосвязи для наборов реализованных элементов программной системы и для вспомогательных средств, а также интерфейсные ограничения.

b) Выполнение интеграции. Последовательно интегрируйте конфигурации элементов программной системы, пока не будет синтезирована вся система. Деятельность состоит из следующих задач:

1) Получение реализованных элементов программной системы в соответствии с согласованными графиками.

Примечание – Реализованные элементы системы программного обеспечения предоставляются разработчиками или поступают от поставщиков, покупателя или других ресурсов и обычно находятся под контролем CM. Элементы обрабатываются в соответствии с соображениями жизнеспособности, безопасности, защиты и конфиденциальности.

2) Интеграция реализованных элементов.

Примечания

1 Задача выполняется для достижения конфигурации элемента системы программного обеспечения (полной или частичной), соединяющей реализованные элементы, как указано в стратегии интеграции, с использованием определенных процедур, описаний управления интерфейсом и соответствующих систем обеспечения интеграции.

2 С точки зрения программного обеспечения, интеграция реализованных элементов может включать соединение частей объектного кода или просто объединение реализованных элементов, которые являются частью конфигурации программного обеспечения, в методическом подходе по частям. Элементы программного обеспечения компилируются в «сборку», чтобы разветвленные блоки были правильно связаны или объединены в собранном элементе. Элементы встроенного программного обеспечения изготавливаются, в виде прототипов, и устанавливаются в аппаратные элементы. Если программные функции еще не доступны для интеграции, эмулированные функции (заглушки или кодогенерации) могут быть использованы для временной поддержки интеграции программных элементов или представления ввода от внешних интерфейсов. Успешное объединение приводит к созданию интегрированного программного элемента, который хранится и доступен для дальнейшей обработки, т.е. дополнительной интеграции, верификации или валидации элемента программной системы.

3 Проблемы защиты от подделок, вскрытия, обеспечения надежности систем и программного обеспечения и совместимости могут возникнуть при выполнении интеграции, выявлении и определении контрольных точек. В процессах интеграции и верификации часто используются фиктивные данные по соображениям безопасности или конфиденциальности. [ISO/IEC/IEEE 15026] и серия [ISO/IEC 27000] содержат информацию о гарантиях, целостности и безопасности, влияющих на интеграцию.

3) Убедитесь, что интегрированные программные интерфейсы или функции работают от запуска и до их планируемого завершения в предполагаемом диапазоне значений данных.

Примечание – В рамках приемки реализованных элементов программной системы, выбранные элементы проверяются, чтобы убедиться, что они соответствуют критериям приемки, указанным в стратегии интеграции и применимых соглашениях. Проверка может включать соответствие согласованной конфигурации, совместимость интерфейсов и наличие обязательных информационных элементов. Процесс оценки и контроля проекта может использоваться в соответствии со стратегией интеграции для планирования и проведения технических обзоров элементов интегрированной программной системы, например, проверки готовности к тестированию, чтобы гарантировать, что интегрированный элемент или система с его аффилированными данными и информационными элементами работает готовы к квалификационным испытаниям.

c) Управление результатами интеграции. Деятельность состоит из следующих задач:

1) Запись результатов интеграции и обнаруженных ошибок.

Примечание – Сюда входят ошибки из-за стратегии интеграции, систем, обеспечивающих интеграцию, выполнения интеграции или неправильного определения системы или элемента. Если существуют несоответствия на интерфейсе между системой, ее указанной операционной средой и системами, обеспечивающими стадию использования, отклонения приводят к корректирующим действиям. Решение ошибок обычно включает технические процессы, часто повторяющееся применение процесса внедрения. Процессы обеспечения качества и оценки и контроля проекта используются для анализа данных с целью определения первопричины, включения корректирующих или улучшающих действий и записи извлеченных уроков.

2) Обеспечение прослеживаемости элементов интегрированной программной системы.

Примечание – Поддерживается двунаправленная прослеживаемость между элементами интегрированной системы и программной системой, архитектурой, дизайном и требованиями к системе или элементам, такими как варианты использования, включая требования и определения интерфейсов, необходимые для интеграции. Интегрированные элементы программного обеспечения и их компоненты идентифицируются по версии. Версии интегрированных программных элементов обычно можно проследить до реализованных модулей, процедур тестирования и тестовых примеров.

3) Предоставление ключевых аспектов и элементов информации, выбранных для базовых показателей.

Примечание – Процесс управления конфигурацией используется для создания и поддержки элементов конфигурации и базовых показателей. Процесс интеграции определяет кандидатов для базового уровня, а процесс управления информацией контролирует информационные элементы. Для этого процесса тестовые примеры, регрессионные тесты и сценарии автоматического тестирования являются типичными аспектами, которые являются базовыми. Стратегия интеграции –типичный элемент информации, имеющий базовую основу.

6.4.9 Процесс верификации

6.4.9.1 Цель

Целью процесса верификации является предоставление объективных свидетельств того, что система или системный элемент удовлетворяет установленным требованиям и характеристикам.

Процесс верификации выявляет ошибок (ошибки, дефекты или сбои) в любом информационном элементе (например, требованиях к системе/программному обеспечению или описании архитектуры), реализованных системных элементах или процессах жизненного цикла с использованием соответствующих методов, приемов, стандартов или правил. Процесс предоставляет необходимую информацию для определения разрешения выявленных ошибок.

Проверка может выполняться по всем техническим процессам. Процесс верификации используется в ключевых точках жизненного цикла программной системы, чтобы продемонстрировать, что требования (включая функциональные и нефункциональные требования) были выполнены, или что результаты процесса были достигнуты, или были выполнены действия процесса. Различные пререгативы и программотехники или сообщества разработчиков могут по-разному определять этапы, стратегии и критерии проверки.

Для программных систем процесс проверки создается для следующих целей:

a) Для подтверждения того, что программный рабочий продукт или услуга должным образом отражает указанные требования (называемое верификацией программного обеспечения);

b) Подтверждение того, что интегрированный программный продукт соответствует определенным требованиям (называемым квалификационным тестированием программного обеспечения);

c) Подтверждение того, что реализация каждого требования к системе/программному обеспечению проверяется на соответствие и что система программного обеспечения готова к поставке (называемое квалификационным тестированием системы).

Примечания

1 Процесс проверки определяет, что «продукт создан правильно». Процесс валидации определяет, что «создан правильный продукт».

2 [ISO/IEC/IEEE 29119] предоставляет подробные процессы и методы проверки, выполняемой посредством тестирования. [IEEE Std 1012-2012] предоставляет дополнительные сведения об этих процессах для систем, программного обеспечения, оборудования и интерфейсов, разрабатываемых, обслуживаемых или повторно используемых.

3 [SWEBOK] содержит подробное обсуждение тестирования программного обеспечения. В этой области знаний рассматриваются основы, терминология, проблемы, методы, применение, планирование процессов, меры, инструменты, практические соображения и ссылки. В руководстве также обсуждается проверка и валидация программного обеспечения с точки зрения процессов управления качеством программного обеспечения и определяются методы и методики, которые поддерживают как верификацию, так и валидацию. [SWEBOK] также затрагивает такие темы, как создание программного обеспечения для проверки и поддержка моделей и методов программной инженерии.

6.4.9.2 Результаты

В результате успешной реализации процесса Верификации:

a) Выявлены ограничения проверки, влияющие на требования, архитектуру или дизайн.

b) Доступны любые вспомогательные системы или услуги, необходимые для проверки.

c) Система или системный элемент проверены.

d) Обобщаются данные, предоставляющие информацию для корректирующих действий.

e) Предоставляется объективное свидетельство того, что реализованная система соответствует требованиям, архитектуре и дизайну.

f) Выявлены результаты поверки и ошибок.

g) Установлена прослеживаемость проверенных элементов системы.

6.4.9.3 Деятельность и задачи

В проекте должны быть реализованы следующие виды деятельности и задачи в соответствии с применимыми политиками и процедурами организации в отношении процесса верификации.

a) Подготовка к проверке. Деятельность состоит из следующих задач:

1) Определение стратегии проверки, которая включает в себя следующее:

Примечания

1 Стратегия проверки направлена на минимизацию затрат, графика или риска, обеспечивая сбалансированный подход для подтверждения того, что программная система или элемент были «построены правильно».

2 Стратегия и расписание проверки учитывают динамические изменения при возникновении ошибок результатов (событий, инцидентов или проблем). В соответствии с ходом выполнения проекта запланированные действия по проверке переопределяются или переносятся в случае возникновения непредвиденных событий или развития системы.

3 Стратегия верификации может быть задокументирована в плане, например, плане верификации, SDP или SEMP проекта.

i) Определение области проверки, включая программную систему, элемент или продукт, проверяемые свойства и ожидаемые результаты.

Примечание – Общая область проверки включает в себя интересующую программную систему или системные элементы, включая интерфейсы. Для каждого действия по верификации указывается программная система, элемент или продукт, подлежащий верификации (например, реальная система или модель, макет, прототип, код, процедура, план или другой документ) и ожидаемые результаты, такие как соответствие или производительность, отказоустойчивость и восстановление после прерывания обслуживания. Проверяемые свойства могут включать требования, архитектурные и проектные характеристики, интеграцию и точность документации. Характеристики проекта могут включать в себя последствия проекта для безопасности в контексте планируемой операционной среды и достижение критических характеристик качества, как указано в требованиях.

ii) Определение ограничений, которые потенциально ограничивают выполнимость действий по проверке.

Примечание – Ограничения включают техническую осуществимость, стоимость, время, наличие средств проверки или квалифицированного персонала, договорные ограничения и такие характеристики, как критичность миссии. Такие ограничения часто влияют на определение стратегии верификации, например, необходимость или оправданность организационно независимых усилий по верификации.

iii) Определение приоритетов проверки.

Примечание – В программных системах проверка всех возможных сценариев (100% покрытие кода) неосуществима. Стратегия проверки обычно включает компромисс между тем, что будет проверяться (область действия), и ограничениями или пределами, а также определение того, какие действия проверки необходимо выполнить и сколько итераций действий проверки и доработки необходимо для снижения риска. Подход к тестированию на основе моделей может позволить создавать и управлять множеством сценариев. Потенциальные действия по проверке, которые являются кандидатами на удаление, оцениваются с точки зрения рисков, связанных с их удалением.

2) Определение ограничений стратегии проверки, которые необходимо включить в требования, архитектуру или дизайн системы/программного обеспечения.

Примечание – Сюда входят практические ограничения точности, неопределенности, повторяемости, которые налагаются инструментами верификации, соответствующими методами измерения, необходимостью интеграции программной системы, а также наличием, доступностью и взаимосвязью с инструментами поддержки.

3) Определение цели, условии и критериев соответствия для каждого действия проверки.

4) Выбор соответствующих методов или методик проверки и соответствующие критерии для действий по проверке, таких как осмотр, анализ, демонстрация или тестирование.

Примечания

1 Выбор методов или приемов верификации производится в соответствии с типом системы, целью верификации, целями проекта и допустимыми рисками. Методы или методы проверки включают осмотр (включая пошаговые руководства по коду и экспертную оценку), анализ (включая моделирование и симуляцию, а также аналогии/сходства), демонстрацию, а также динамическое и статическое тестирование.

2 Выбранные подход, методы и приемы проверки могут быть согласованы с соответствующими заинтересованными сторонами, чтобы обеспечить приемлемость подхода к проверке.

5) Определение и планирование необходимых вспомогательных систем или услуг, необходимых для поддержки проверки.

Примечание – Системы обеспечения верификации включают средства верификации, квалифицированный персонал, оборудование, симуляторы, инструменты автоматизации тестирования, а также системы управления инцидентами и проблемами. Верификация программной системы обычно выполняется в отдельных контролируемых средах, которые не мешают работе программного обеспечения или текущей разработке. Если вспомогательные системы для проверки отличаются по возможностям от запланированной операционной среды, процесс измерения может быть использован для калибровки производительности вспомогательных систем проверки и их пригодности для действия проверки.

6) Получение или приобретение доступа к вспомогательным системам или услугам, которые будут использоваться для поддержки проверки.

Примечание – Приобретение вспомогательных систем может осуществляться различными способами, такими как аренда, закупка, разработка, повторное использование активов организации или субподряд. Получение полного набора вспомогательных средств представляет собой сочетание этих способов. Процесс валидации используется для объективного подтверждения того, что разрешающая система верификации достигает своего предполагаемого использования для своих разрешающих функций.

b) Выполнение проверки. Эта деятельность состоит из следующих задач:

1) Определить процедуры проверки, каждая из которых поддерживает одно или несколько проверочных действий.

Примечание – Процедуры проверки, которые могут выполняться с помощью автоматизированных сценариев, включают в себя требования, подлежащие проверке, тип элемента программной системы или продукта, подлежащего проверке (например, реальная система или модель, макет, прототип, код, процедура, план или другой элемент информации), а также ожидаемые результаты (критерии успеха), такие как соответствие или производительность функции или мощности с точки зрения времени отклика или пропускной способности. В процедурах указывается цель проверки с критериями успеха (ожидаемые результаты), применяемая техника проверки, необходимые вспомогательные системы (помещения, оборудование) и условия окружающей среды для выполнения каждой процедуры проверки (ресурсы, квалифицированный персонал, специализированные процедурные инструкции по установке или работе).

2) Выполнить процедуры проверки.

Примечание – Верификация происходит в соответствии со стратегией верификации в соответствующее время по расписанию, в определенной среде, с определенными вспомогательными системами и ресурсами. Выполнение действия проверки состоит из фиксации результата выполнения процедуры проверки; сравнение полученного и зафиксированного результата с ожидаемым результатом; и определение степени правильности (или успеха/неудачи) представленного элемента.

c) Управление результатами проверки. Деятельность состоит из следующих задач:

1) Просмотр результатов проверки и обнаруженных аномалий и определение дальнейших действий.

Примечания

1 Сюда входят аномалии из-за стратегии проверки, систем включения проверки, выполнения проверки или неправильного определения системы. Процессы оценки и контроля проекта и обеспечения качества используются для анализа данных с целью определения первопричины, принятия корректирующих или улучшающих действий, а также для записи извлеченных уроков.

2 Оценка результатов проверки и последующие корректирующие действия могут сильно различаться в зависимости от цели проверки. Для элементов программного обеспечения примеры включают модификацию или отказ от требований, простое исправление дефекта для отказавшего элемента программного обеспечения с последующей повторной проверкой или изменение направления крупного проекта на основании неспособности достичь ключевой вехи, например, отказавшего программного обеспечения. система квалификационного тестирования. Часто простые или рекомендуемые решения ошибок, обнаруженных во время проверки, записываются вместе с результатами проверки, чтобы облегчить анализ и возможные корректирующие действия.

2) Запись инцидентов и проблем во время проверки и отслеживание их решения.

Примечание – Решение проблем осуществляется посредством процессов обеспечения качества и оценки и контроля проекта. Во время проверки программного обеспечения документируются условия, при которых возникла проблема, чтобы по возможности можно было продублировать проблему и определить основную причину дефекта программного обеспечения. Изменения требований, архитектуры, дизайна или элементов системы выполняются с использованием других технических процессов.

3) Получение согласия заинтересованных сторон, что программная система или элемент соответствует указанным требованиям.

4) Поддержание отслеживаемости проверенных элементов программной системы.

Примечание – Поддерживается двунаправленная прослеживаемость между проверенными элементами системы и записями о деятельности по верификации, системной архитектуре, дизайне или системным/программным требованиям.

5) Предоставление ключевых ошибок и элементов информации, выбранных для базовых показателей.

Примечание – Процесс управления конфигурацией используется для создания и поддержки элементов конфигурации и базовых показателей. Процесс определяет кандидатов для базового плана, а процесс управления информацией контролирует информационные единицы. Для этого процесса стратегия проверки и процедуры проверки являются типичными информационными элементами.

6.4.10 Процесс перехода

6.4.10.1 Цель

Целью процесса перехода является создание способности системы предоставлять услуги, определенные требованиями заинтересованных сторон в операционной среде.

Процесс переводит систему упорядоченным, запланированным образом в рабочий статус, так что система является функциональной, работоспособной и совместимой с другими операционными системами. Он устанавливает проверенную систему вместе с соответствующими вспомогательными системами, например, системой планирования, системой поддержки, системой обучения операторов, системой обучения пользователей, как определено в соглашениях. Этот процесс используется на каждом уровне структуры системы и на каждом этапе для выполнения критериев, установленных для выхода из этапа. Он включает в себя подготовку соответствующих систем хранения, погрузочно-разгрузочных работ и отгрузки.

Для программных систем цель процесса перехода – установить способность системы предоставлять услуги в другой среде.

Процесс перехода используется для повторяющихся развертываний программного обеспечения в различных средах, например, из среды разработки в среду тестирования или обслуживания, или между различными тестовыми средами, или из одной операционной среды в другую (например, повторное размещение или использование облачных услуг). Переход на резервные или условные площадки планируется и отрабатывается для обеспечения непрерывности процесса и аварийного восстановления. Переход для программных систем может включать физическое перемещение оборудования, установку и активацию или деактивацию физической или виртуальной инфраструктуры или вспомогательных систем в разных местах, или отсутствие изменений в физической инфраструктуре. Переход может включать изменения в ресурсах данных, структуре данных, обновления или модернизации функционального программного обеспечения. Переход включает повторяющиеся плановые или аварийные исправления, а также исправления безопасности и исправления других проблем. Переход может включать в себя передачу данных между организациями, а также добавление большой группы новых пользователей к существующей программной системе или услуге. Переход на новую систему осуществляется одновременно с выводом из эксплуатации и утилизацией существующей системы, что влечет за собой миграцию данных из старой системы в ее замену.

Примечание – Переход может включать передачу знаний с использованием процесса управления знаниями.

6.4.10.2 Результаты

В результате успешной реализации процесса перехода:

a) Выявлены ограничения перехода, которые влияют на системные / программные требования, архитектуру или дизайн.

b) Доступны любые вспомогательные системы или услуги, необходимые для перехода.

c) Сайт подготовлен.

d) Система, установленная на рабочем месте, способна выполнять указанные функции.

e) Операторы, пользователи и другие заинтересованные стороны, необходимые для использования и поддержки системы, проходят обучение.

f) Выявлены результаты перехода и ошибок.

g) Установленная система активирована и готова к работе.

h) Установлена прослеживаемость переходных элементов.

6.4.10.3 Деятельность и задачи

В рамках проекта должны быть реализованы следующие виды деятельности и задачи в соответствии с применимыми политиками и процедурами организации в отношении процесса перехода.

a) Подготовка к переходу системы программного обеспечения. Эта деятельность состоит из следующих задач:

1) Определение стратегии управления выпусками программного обеспечения и переходами других программных систем, включая следующие соображения:

i) установление типа перехода и критериев успеха перехода;

ii) определение частоты повторяющихся переходов, таких как обновления и обновления систем программного обеспечения для разработки, тестирования и эксплуатации;

iii) минимизация рисков безопасности, сбоев и простоев во время перехода;

iv) архивирование, уничтожение или преобразование и проверка данных из предыдущих систем в новую систему, в том числе данные, полученные через внешние интерфейсы;

v) планирование на случай непредвиденных обстоятельств для решения проблемы, резервного копирования и возврата к последней рабочей версии системы;

vi) планирование переходов в соответствии с текущими бизнес-процессами, с поэтапным или синхронным переходом систем;

vii) управление изменениями для заинтересованных сторон, включая партнеров по интерфейсу, людей-операторов, системных администраторов и пользователей программных систем или служб;

Примечание – Действия по управлению изменениями проводятся для разработки изменений в бизнес-процессах, связанных с новой системой, планирования перехода в бизнес-процессах и обеспечения приверженности пользователей к продуктивному использованию новой системы.

viii) связанные стратегии для проверки переходной системы или элемента;

ix) инициирование действий по поддержке и обслуживанию пользователей с передачей и обновлением проектной документации системы, пользовательской документации и процедур тестирования;

x) одновременное выполнение процессов перехода, операций и утилизации, когда новая система вводится в эксплуатацию, а старая система выводится из эксплуатации.

Примечание – Стратегия включает роли и обязанности, полномочия по утверждению, использование проверок готовности и обучение.

2) Выявление и определение изменения объекта, площадки, сети связи или целевой среды, необходимые для установки или перехода на новую систему программного обеспечения.

Примечание – Для каждого перехода идентифицируйте и определите любые необходимые изменения в инфраструктуре или вспомогательных системах. Обследование объекта может быть выполнено для выявления необходимых изменений в физической среде для установки или использования системы программного обеспечения, таких как изменения для поддержания физической и информационной безопасности системы.

3) Определение информационных потребностей и организация пользовательской документации и обучение операторов, пользователей и других заинтересованных сторон, необходимых для использования и поддержки системы.

Примечание – Переход включает в себя миграцию или активацию доступа пользователя к программной системе. Установлены роли пользователей, реализованы учетные записи пользователей и средства контроля доступа.

4) Подготовка подробной информации о переходе, такой как планы, графики и процедуры.

Примечания

1 Стратегия перехода обычно записывается в плане, например, плане перехода или SDP или SEMP проекта. Графики перехода помогают проверить наличие достаточных ресурсов и инфраструктуры для поддержки перехода, так что действия могут быть выполнены в установленные сроки, чтобы свести к минимуму сбои. Расписание может включать тестирование для сложных переходов, в которых процедуры, такие как резервное копирование и восстановление базы данных и системы, а также установка программного обеспечения, тестируются для проверки продолжительности и правильности результатов.

2 В течение определенного периода переключения или одновременной работы передача услуг управляется таким образом, чтобы постоянное соответствие постоянным потребностям заинтересованных сторон или согласованному уровню обслуживания достигалось. Если требуется период параллельных операций, как для старой, так и для новой системы, определяются и разрабатываются специальные процедуры для получения и использования данных от партнеров по интерфейсу.

5) Определение системных ограничений от перехода до включения в системные требования, архитектуру или дизайн программного обеспечения.

6) Определение и планирование необходимых вспомогательных систем или услуг, необходимых для поддержки перехода.

Примечание – Включает определение требований и интерфейсов для вспомогательных систем. Переход включает использование высокоавтоматизированной инфраструктуры для доставки, установки, активации или деактивации программного обеспечения. Для электронного распространения программного обеспечения требуются временные или постоянные изменения в подключении для миграции программного обеспечения и данных и постоянной поддержки. Разрешающие системы могут включать резервные или альтернативные системы для использования в переходный период.

7) Получение или приобретение доступа к вспомогательным системам или услугам, которые будут использоваться.

Примечание – Процесс валидации используется для объективного подтверждения того, что система, разрешающая переход, достигает своего предполагаемого использования для своих разрешающих функций.

b) Выполнение перехода. Деятельность состоит из следующих задач:

1) Подготовка места эксплуатации или виртуальной среды в соответствии с требованиями установки.

Примечание – Подготовка площадки проводится в соответствии с применимыми нормативными актами в области здравоохранения, безопасности и охраны окружающей среды. Инициализируются и проверяются виртуальные среды и новые коммуникационные ресурсы. Организована отгрузка и прием физических элементов системы и вспомогательных систем.

2) Доставка программной системы или элемента для установки в необходимое место и время.

Примечания

1 Программное обеспечение поставляется в электронном виде. Для физических носителей, оборудования и встроенных программных систем необходимо учитывать временное хранилище перед доставкой или установкой.

2 Программное обеспечение поставляется с согласованными информационными элементами в электронной или бумажной форме, такие как учебные материалы, пакеты логистической поддержки или пользовательская документация.

3) Установка продукта в его физическом или виртуальном рабочем месте и подключение к его среде.

Примечание – Установка продукта включает настройку необходимых операционных данных, изменений среды или бизнес-процессов Базы данных инстанцируются, и при необходимости выполняется миграция данных. Лицензии и соглашения об обслуживании элементов системы и другой интеллектуальной собственности передаются в соответствии с соглашениями.

4) Предоставление пользовательской документации и обучение операторов, пользователей и других заинтересованных сторон, необходимых для использования и поддержки продукта.

5) Выполнение активации и выписки, включая следующее по согласованию.

Примечания

1 В этой задаче предпринимаются шаги, необходимые для активации продукта до рабочего состояния, включая запуск, оценку условий окружающей среды и другие оценки готовности, в соответствии с операционными процедурами, политиками организации и правилами. В тех случаях, когда точное местоположение или среда работы недоступны или когда к программному обеспечению будет осуществляться доступ из нескольких или мобильных мест, выбирается типичный пример.

2 Приемочные испытания определяются в соглашении, чтобы продемонстрировать удовлетворительную установку. Эта задача взаимодействует с процессом проверки, чтобы объективно подтвердить, что система выполняет требования заинтересованных сторон в операционной среде. Приемочные испытания, как указано в соглашениях, могут определять критерии, которые демонстрируют, что объект программной системы обладает способностью предоставлять требуемые функции и услуги при установке и поддержании в своей операционной среде. Особое внимание уделяется ключевым функциям и логическим интерфейсам.

3 Как часть процесса управления конфигурацией, аудит физической конфигурации (PCA) и обновление исполнительной документации часто выполняется во время активации системы. Могут быть подтверждены положения о борьбе с контрафактом.

i) Демонстрация правильной установки программного обеспечения.

Примечание – Задача может включать в себя проверки целостности данных и операций, например, что программный код и представления данных должным образом инициализируются, выполняются и завершаются, как указано.

ii) Демонстрация установленного или переведенного продукта на способность выполнять требуемые функции.

Примечание – Задача эксплуатационной готовности, которая проверяет готовность функциональных возможностей к рабочему состоянию. Особое внимание уделяется интерфейсам данных и проблемам безопасности: выполняются функции обеспечения информации и взаимодействия.

iii) Демонстрация функции, выполняемые системой, на устойчивость с помощью вспомогательных систем.

Примечание – Задача эксплуатационной готовности, которая проверяет готовность активирующих систем к рабочему состоянию. Например, демонстрируются активация мониторинга, создание отчетов о проблемах, контроль доступа, резервное копирование и восстановление, а также помощь пользователям (поддержка клиентов).

iv) Проверка программной системы на готовность к работе.

Примечание – Сюда входят результаты функциональных демонстраций, валидации и демонстрации устойчивости. Может быть проведена проверка готовности. Недостатки, риски и проблемы, влияющие на успех перехода, устраняются, принимаются для отказа или закрываются.

v) Введение программного комплекса в эксплуатацию.

Примечание – Сюда входит оказание поддержки пользователям, администраторам и операторам во время ввода в эксплуатацию (ввода в эксплуатацию) системы.

c) Управление результатами перехода. Деятельность состоит из следующих задач:

1) Запись результатов перехода и обнаруженных аномалий.

Примечание – Сюда входят аномалии из-за стратегии перехода, систем включения перехода, выполнения перехода или неправильного определения системы программного обеспечения или системы баз данных. Если существуют несоответствия между системой, ее операционной средой и вспомогательными системами, отклонения устраняются с помощью корректирующих действий, включая изменения требований. Процессы оценки и контроля проекта и обеспечения качества используются для анализа данных с целью выявления первопричины, включения корректирующих или улучшающих действий и записи извлеченных уроков.

2) Запись инцидентов и проблем перехода и отслеживайте их решение.

Примечание – Решение проблем осуществляется посредством процессов обеспечения качества и оценки и контроля проекта. Во время перехода условия, при которых возникла проблема, документируются, чтобы по возможности можно было продублировать проблему и определить первопричину дефекта. Изменения требований, архитектуры, дизайна или элементов программной системы выполняются с использованием других технических процессов.

3) Поддержка отслеживаемости переданных элементов программной системы.

Примечание – Поддерживается двунаправленная прослеживаемость между перенесенной и развернутой системой и элементами, а также утвержденными и контролируемыми версиями программной системы и вспомогательных систем.

4) Предоставление ключевых аспектов и элементов информации, выбранные для базовых показателей.

Примечание – Процесс управления конфигурацией используется для создания и поддержки элементов конфигурации и базовых показателей, в том числе переданных элементов программной системы. Этот процесс определяет кандидатов для базового уровня, а процесс управления информацией контролирует информационные единицы. Для этого процесса стратегия перехода, учебные материалы, процедуры установки, перехода и миграции данных, а также пользовательская документация являются типичными информационными элементами, которые являются базовыми.

6.4.11 Процесс валидации

6.4.11.1 Цель

Целью процесса валидации является предоставление объективных свидетельств того, что система, когда она используется, выполняет свои бизнес-задачи или задачи и требования заинтересованных сторон, достигая своего предполагаемого использования в предполагаемой операционной среде.

Целью валидации системы или системного элемента является получение уверенности в их способности выполнять намеченную миссию или использовать в конкретных условиях эксплуатации. Валидация утверждается заинтересованными сторонами. Процесс предоставляет необходимую информацию, чтобы выявленные ошибки могли быть устранены с помощью соответствующего технического процесса, в котором ошибка была создана.

Процесс валидации обычно используется в ключевых точках жизненного цикла продукта, чтобы продемонстрировать, что требования продукта для предполагаемого эксплуатационного использования заинтересованными сторонами выполнены. Валидация также применима к ошибкам программной инженерии (рассматриваемым как элементы программной системы). Различные пререгативы и программотехники или сообщества разработчиков могут по-разному определять этапы, стратегии и критерии проверки.

Для программных систем модели жизненного цикла с высокой итерацией включают участие покупателя, представителя пользователя или других заинтересованных сторон для проверки, например, приоритета требований для включения в итерацию, удобства использования программного интерфейса через прототипы и пригодности программного обеспечения для выполнения бизнес-задач и реализации операционной концепции.

Для программных систем цели процесса валидации:

a) Подтверждение того, что требования к конкретному предполагаемому использованию рабочего продукта программного обеспечения выполнены (называется валидацией программного обеспечения);

b) Достижение уверенности (особенно у приобретателя или пользователя) в том, что поставленный продукт соответствует требованиям заинтересованных сторон и пригоден для использования (это часто называется приемочным тестированием программного обеспечения).

Примечания

1 Процесс проверки определяет, что «правильный продукт создан». Процесс проверки определяет, что «продукт создан правильно».

2 Критерии приемки, используемые для приемочных испытаний, включают критерии для определения того, пригоден ли поставляемый продукт для использования или нет. Критерии приемки для приемки могут быть определены и согласованы между двумя сторонами, т. е. покупателем и поставщиком, и включены в требования заинтересованных сторон.

3 [IEEE Std 1012-2012] содержит подробные требования. [SWEBOK] обсуждает верификацию и валидацию программного обеспечения с точки зрения процессов управления качеством программного обеспечения и содержит методы и приемы, которые поддерживают как верификацию, так и валидацию. [SWEBOK] также затрагивает такие темы, как требования и проверка модели.

6.4.11.2 Результаты

В результате успешной реализации процесса валидации:

a) Определены критерии проверки требований заинтересованных сторон.

b) Подтверждена доступность услуг, требуемых заинтересованными сторонами.

c) Выявлены ограничения валидации, влияющие на требования, архитектуру или дизайн.

d) Система или системный элемент проверены.

e) Доступны любые вспомогательные системы или услуги, необходимые для проверки.

f) Выявлены результаты валидации и ошибки.

g) Предоставляется объективное свидетельство того, что реализованная система или системный элемент удовлетворяет потребности заинтересованных сторон.

h) Установлена прослеживаемость утвержденных элементов системы.

6.4.11.3 Деятельность и задачи

В рамках проекта должны быть реализованы следующие виды деятельности и задачи в соответствии с применимыми политиками и процедурами организации в отношении процесса валидации.

a) Подготовка к проверке. Деятельность состоит из следующих задач:

1) Определение стратегии проверки, которая включает в себя следующее:

Примечания

1 Стратегия валидации направлена на минимизацию стоимости, графика или риска путем постепенного укрепления уверенности в качестве и пригодности программной системы для заинтересованных сторон.

2 Стратегия валидации отражает модель жизненного цикла и включает повторную валидацию для итеративных, инкрементальных или эволюционных жизненных циклов.

3 Стратегия валидации может быть задокументирована в плане, например, в плане приемки, или в SDP или SEMP проекта.

i) Определение области валидации, включая характеристики программной системы, элемента или ошибки, подлежащего валидации, и ожидаемые результаты валидации.

Примечание – Валидация программных систем проводится как в отдельных контролируемых средах, которые не влияют на работу программного обеспечения или текущей разработке, так и в рабочих средах, перед полным эксплуатационным использованием (например, бета-тестирование или приемочные испытания в течение определенного срока с согласованными критериями). Объем включает требования заинтересованных сторон, в том числе соответствующие представления о системе (например, варианты развития ситуации или концепции эксплуатации), подлежащие оценке. Объем зависит от того, что подходит для этапа жизненного цикла системы: интересующая система или элемент системы или инженерный продукт, такой как описание концепции или документ, условия эксплуатации, модель, макет или прототип. Область применения также включает оценку того, что программный продукт или услуга пригодны к использованию в предназначенной для них среде для выполнения основных или критических функций. Дополнительные характеристики, подлежащие проверке, могут включать удобство использования документации; отказоустойчивость, устойчивость и возможности восстановления программного обеспечения.

ii) Определение ограничений, которые потенциально ограничивают выполнимость действий по проверке.

Примечание – Ограничения включают практические ограничения точности, неопределенности, повторяемости, на которые накладываются средства поддержки валидации, соответствующие методы измерения, а также наличие, доступность и взаимосвязь со средствами поддержки. Стратегия валидации ограничена ходом проекта; в частности, запланированные действия по валидации переопределяются или переносятся при возникновении неожиданных событий или развития системы. Валидацию можно расширить, включив в нее текущие измерения удовлетворенности пользователей и жалоб клиентов.

iii) Определение приоритетов проверки.

Примечания

1 Для эффективного использования времени и знаний заинтересованных сторон, валидация основывается на приоритетах заинтересованных сторон, в то время как верификация используется для нефункциональных требований. Потенциальные действия по валидации, которые являются кандидатами на удаление, оцениваются с точки зрения рисков, связанных с их отменой.

2 Поставщик, покупатель или агент покупателя участвует в проверке достоверности или выполняет ее. Ответственность часто указывается в соглашении.

2) Определение системных ограничений из стратегии валидации, которые будут включены в требования заинтересованных сторон.

3) Определение цели, условия и критерия соответствия для каждого действия по валидации.

4) Выбор соответствующих методы или методик проверки и соответствующие критерии для каждого действия по проверке.

Примечания

1 Методы или приемы валидации программных систем включают проверку, анализ, аналогию/подобие, демонстрацию, моделирование, экспертную оценку и тестирование. Методы валидации программного обеспечения включают демонстрации, проверки, обзоры и пошаговые инструкции по коду, тесты удобства использования и пробное использование программного обеспечения (например, бета-тестирование, эксплуатационное тестирование, пользовательское тестирование или приемочное тестирование с согласованными критериями). Выбор методов или приемов валидации осуществляется в соответствии с типом системы, целью валидации, целями проекта, законодательными и нормативными требованиями и приемлемыми рисками валидационного действия. Для программных систем взамодействующих с оператором, удобство использования тестирование используется для проверки того, что типичные пользователи могут достичь определенных целей с результативностью, эффективностью и удовлетворенностью в конкретном контексте использования. Дополнительные сведения о тестировании удобства использования приведены в [ISO/IEC TR 25060: 2010].

2 В соответствующих случаях определяются этапы или состояния валидации (например, валидация внутри организации, валидация на месте, эксплуатационная валидация), которые постепенно повышают уверенность в соответствии развивающейся программной системы и помогают диагностировать любые обнаруженные несоответствия.

3 Критерии принятия заинтересованными сторонами качества обслуживания указываются как уровень обслуживания и записываются в соглашении об уровне обслуживания (SLA). Уровни обслуживания измеряют емкость, доступность, надежность и своевременное реагирование на услуги и приводят к требованиям к производительности для поддерживающих систем.

5) Определение и планирование необходимых вспомогательных систем или услуг для поддержки валидации.

Примечание – Вспомогательные системы могут включать тренажеры, лабораторию или испытательную лабораторию, квалифицированный персонал, представителей заинтересованных сторон и пользователей в соответствии с выбранными методами или методами проверки. Включает определение требований и интерфейсов для вспомогательных систем.

6) Получить или приобрести доступ к вспомогательным системам или услугам, которые будут использоваться для поддержки валидации.

Примечание – Процесс управления инфраструктурой или процесс приобретения может быть вызван для получения доступа к поддерживающим системам, например, через аренду, закупку, разработку, повторное использование или субподряд. Доступ к полному набору средств поддержки представляет собой сочетание этих способов. Процесс валидации используется для объективного подтверждения того, что разрешающая система валидации достигает своего предполагаемого использования для своих разрешающих функций.

b) Выполнить проверку. Деятельность состоит из следующих задач:

1) Определение процедуры проверки, каждая из которых поддерживает одно или несколько действий проверки.

Примечание – Процедуры валидации идентифицируют требования заинтересованных сторон, подлежащие валидации, связанный с ними ошибки программной системы (например, реальная система или модель, макет, прототип, код, набор инструкций или другой информационный элемент) и ожидаемые результаты (критерии успеха), такие как завершенное и своевременное выполнение функции. В процедурах указывается цель валидации с критериями успеха (ожидаемые результаты), применяемый метод валидации, необходимые вспомогательные системы (помещения, оборудование) и условия окружающей среды для выполнения каждой процедуры валидации (ресурсы, квалифицированный персонал, участвующие заинтересованные стороны). А также специализированные процедурные настройки или рабочие инструкции). Стратегия проверки включает в себя то, как результаты процедуры проверки будут записываться, анализироваться, храниться и сообщаться.

2) Выполнение процедуры проверки в определенной среде.

Примечание – Валидация происходит в соответствии со стратегией валидации в соответствующее время в расписании, в определенной среде (такой как операционная среда, аналогичная тестовая среда или другая репрезентативная среда) с определенными инструментами поддержки и ресурсами. Выполнение действия проверки обычно состоит из сбора результатов выполнения, сравнения полученного результата с критериями успеха и определения степени соответствия или удовлетворенности заинтересованных сторон программной системой, элементом, услугой или инженерным продуктом.

c) Управление результатами проверки. Деятельность состоит из следующих задач:

1) Просмотр результатов проверки и обнаруженных ошибок и определение дальнейших действий.

Примечания

1 Необходимо подтверждение, что сервисы системы, которые требуются заинтересованным сторонам, доступны. Ошибки могут быть результатом стратегии валидации, систем, обеспечивающих валидацию, выполнения валидации, неправильного определения системы или неэффективного или неэффективного проектирования, внедрения и интеграции системы.

2 Оценка результатов валидации и последующие действия могут включать принятие ошибок как происшествия с низким уровнем риска. Корректирующие действия могут сильно различаться в зависимости от результата проверки. Для элементов программного обеспечения примеры включают простое исправление дефекта для отказавшего элемента программного обеспечения, дополнительное обучение пользователей, исправления и пояснения в документации или изменение направления крупного проекта на основе неспособности достичь ключевого этапа, например, неудачной приемки программной системы тестирование.

2) Запись инцидентов и проблем во время проверки и отслеживание их разрешения.

Примечание – Процесс оценки и контроля проекта и процесс обеспечения качества используются для анализа данных с целью выявления первопричины проблем, включения корректирующих или улучшающих действий, а также для записи извлеченных уроков. Во время валидации программного обеспечения документируется разрыв между ожиданиями заинтересованных сторон и производительностью системы, чтобы по возможности можно было определить основную причину несоответствия. Решение проблемы включает определение серьезности и воздействия проблемы, а также необходимость исправления или принятия на какое-то время несоответствия программного обеспечения как известной ошибки. Часто простые или рекомендуемые решения аномалий, обнаруженных во время валидации, записываются вместе с результатом валидации, чтобы облегчить анализ и возможные корректирующие действия. Фактические изменения заинтересованных сторон и требований к системе/программному обеспечению, архитектуре, дизайну или элементам системы выполняются в рамках других технических процессов.

3) Получение согласия заинтересованных сторон о том, что программная система или элемент соответствует потребностям заинтересованных сторон.

4) Поддержка отслеживаемости проверенных элементов системы.

Примечание – Поддерживается двунаправленная прослеживаемость между проверенными элементами системы и требованиями заинтересованных сторон и записью результатов валидации.

5) Предоставление ключевых ошибок и элементов информации, выбранных для базовых показателей.

Примечание – Процесс управления конфигурацией используется для создания и поддержки элементов конфигурации и базовых показателей. Этот процесс определяет кандидатов для базовой линии (например, утвержденную систему или элемент программного обеспечения), а процесс управления информацией контролирует информационные единицы. Для этого процесса стратегия валидации и результаты валидации являются типичными базовыми элементами информации.

6.4.12 Операционный процесс

6.4.12.1 Цель

Целью процесса эксплуатации является использование системы для предоставления своих услуг.

Процесс устанавливает требования и назначает персонал для работы с системой, а также контролирует услуги и производительность системы оператора. Чтобы поддерживать услуги, он выявляет и анализирует операционные ошибки в отношении соглашений, требований заинтересованных сторон и организационных ограничений.

Процесс эксплуатации направлен на контроль или снижение стоимости операций при поддержании приемлемого или улучшенного уровня обслуживания.

Программные системы могут иметь выделенную инфраструктуру, но обычно они работают в распределенной среде, где активны другие программные системы и сервисы (например, интернет). Таким образом, безопасность, доступность и эксплуатационные характеристики интересующей программной системы являются предметом заботы в рамках более крупной суперсистемы. Это может включать координацию с уже существующими, параллельными или продолжающимися услугами, предоставляемыми другими системами, которые предоставляют идентичные или подобные услуги.

Примечание – [ISO/IEC 20000-1: 2011 (IEEE Std 20000-1)] – стандарт системы управления услугами, который определяет требования к проектированию, переходу, доставке и улучшению управляемых операционных услуг, а также поддерживает процесс эксплуатации для достижения своей цели.

6.4.12.2 Результаты

В результате успешного выполнения операционного процесса:

a) Идентифицируются операционные ограничения, которые влияют на системные / программные требования, архитектуру или дизайн.

b) Доступны любые вспомогательные системы, услуги и материалы, необходимые для работы.

c) Доступны обученные, квалифицированные операторы.

d) Предоставляются услуги системных продуктов, отвечающие требованиям заинтересованных сторон.

e) Производительность системного продукта во время работы контролируется.

f) Оказывается поддержка заказчику.

6.4.12.3 Деятельность и задачи

В рамках проекта должны быть реализованы следующие виды деятельности и задачи в соответствии с применимыми политиками и процедурами организации в отношении Операционного процесса.

a) Подготовка к работе. Эта деятельность состоит из следующих задач:

1) Определение стратегии работы, включая следующие цели:

i) Ожидаемая или согласованная пропускная способность, доступность, время отклика и безопасность услуг при их внедрении, регулярной эксплуатации и прекращении обслуживания;

ii) Стратегия человеческих ресурсов, в зависимости от необходимости определить требования к обучению и квалификации, обучить или нанять персонал для контроля и мониторинга операций системы программного обеспечения, администрирования доступа к системе и поддержки запросов на обслуживание клиентов и помощи пользователям;

iii) Критерии и графики выпуска программной системы, позволяющие вносить изменения, поддерживающие существующие или расширенные услуги;

iv) Подход к реализации режимов работы в Операционной концепции, включая нормальные операции, а также подготовку и тестирование предусмотренных типов операций в чрезвычайных ситуациях;

v) Меры по эксплуатации, которые обеспечат понимание уровней производительности;

vi) Стратегия эксплуатационной безопасности и охраны труда для операторов и других лиц, использующих программную систему или находящихся в контакте с ней во время работы, с учетом правил техники безопасности;

vii) Стратегия защиты окружающей среды и устойчивого развития для работы программной системы.

Примечание – [ISO/IEC 16350] содержит руководство по эксплуатационным аспектам.

2) Определение системных ограничений в работе, которые будут включены в изменения требований к системе/программному обеспечению, архитектуре, дизайну, реализации или переходу.

3) Определение и планирование необходимых вспомогательных систем или услуг для поддержки работы.

Примечание – Сюда входит определение эксплуатационных требований и интерфейсов для вспомогательных систем. Специальные режимы операционной системы программного обеспечения, например, режим обучения, могут быть активными вместе с полным рабочим режимом или вместо него. Разрешающие системы включают мониторинг изменений угроз программной системе.

4) Получение или приобретение доступа к вспомогательным системам или услугам, которые будут использоваться.

Примечание – Процесс валидации используется для объективного подтверждения того, что разрешающая система функционирует по назначению для своих разрешающих функций.

5) Выявление или определение требований к обучению и квалификации персонала, необходимого для работы программной системы.

Примечание – Обучение и квалификация включает осведомленность о программной системе в ее операционной среде и определенную программу ознакомления с соответствующими инструкциями по обнаружению и устранению неисправностей. Требования к знаниям, навыкам и опыту оператора определяют критерии отбора персонала, и, в соответствующих случаях, подтверждается их разрешение на работу. Объем квалификации зависит от интересующей системы и ее среды. Например, в некоторых средах нормативные требования включают сертификацию операторов, тогда как в других требования сертификации отсутствуют.

6) В зависимости от необходимости вмешательства человека и контроля операций, назначение обученного, квалифицированного персонала операторами.

Примечание – С должным учетом разделения обязанностей, таких как административный контроль доступа к системе и исследование проблем безопасности, многие современные программные продукты сводят к минимуму потребность в операторах, в отличие от конечных пользователей. Операторы обычно поддерживают вспомогательные системы, такие как облачные сервисы, базы данных и системное программное обеспечение, мониторы безопасности, хранилище данных и службу поддержки.

b) Выполнение операции. Деятельность состоит из следующих задач:

1) Использование программной системы в предполагаемой операционной среде.

Примечание – Там, где это согласовано, поддерживается непрерывный потенциал и качество услуг, когда программная система заменяет существующую систему или элемент, который выводится из эксплуатации..

2) При необходимости использование материалов и других ресурсов для работы системы программного обеспечения и поддержки ее услуг.

Примечание – Сюда входят энергоресурсы для оборудования, возможности подключения для программного обеспечения, а также операторов или автоматизированных операторов.

3) Мониторинг работы программной системы, в том числе с учетом следующего:

i) Управление соблюдением операционной стратегии (например, операционных процедур);

ii) Запись и уведомление о значительных событиях, таких как возможные нарушения конфиденциальности и целостности программного обеспечения и данных;

iii) Безопасное использование системы программного обеспечения в соответствии с законодательными нормами, например, в отношении безопасности труда и защиты окружающей среды;

iv) Запись, когда производительность программной системы или услуги не соответствует допустимым параметрам.

Примечание – Сюда входят ошибки из-за стратегии работы, систем включения операции, выполнения операции или неправильного определения системы программного обеспечения. Система демонстрирует неприемлемую производительность, когда системные элементы, реализованные в аппаратном обеспечении, ухудшились или превысили свой срок службы или операционная среда системы влияет на работу программного обеспечения, например, рабочая нагрузка превышает пороговые значения мощности, использовались конкурирующие приложения, взломы безопасности или программные дефекты.

4) В соответствии с операционной стратегией, разработка и, где это возможно, автоматизация операционных процедурхы, чтобы минимизировать риск операционных ошибок.

Примечание – Сюда входят процедуры обработки стандартных (предварительно утвержденных) запросов на изменение и сервисных запросов, устранения неполадок и отчетности об инцидентах, особенно для инцидентов безопасности.

5) В соответствии с операционной стратегией анализ измерения, чтобы подтвердить, что:

i) Производительность услуг находится в пределах приемлемых параметров или согласованных уровней обслуживания для согласованной рабочей нагрузки;

ii) Доступность системы и сервиса и время отклика приемлемы;

iii) Стоимость эксплуатации соответствует целям и ограничениям;

iv) Выявлены и приоритизированы потенциальные улучшения.

Примечание – Отзывы и предложения оператора являются полезным вкладом для улучшения рабочих характеристик программной системы. Могут применяться процессы обеспечения качества и измерения.

6) При необходимости выполнение действий на случай непредвиденных обстоятельств.

Примечание – Включает в себя работу программной системы в ухудшенном режиме, выполнение операции возврата и восстановления, выключение системы, выполнение процедур обхода для восстановления работы или другие режимы для особых условий. При необходимости оператор выполняет действия, необходимые для начала действий в чрезвычайных ситуациях и, возможно, отключения системы. Операции на случай непредвиденных обстоятельств выполняются в соответствии с заранее установленными процедурами для такого события. Часто эти процедуры сопровождаются планом непрерывности.

c) Управление результатами операции. Деятельность состоит из следующих задач:

1) Запись результатов работы и обнаруженных ошибок.

Примечание – Процессы оценки и контроля проекта и обеспечения качества используются для анализа данных об инцидентах и проблемах с целью определения первопричины, включения корректирующих или улучшающих действий, а также для записи извлеченных уроков.

2) Запись производственных инцидентов и проблем и отслеживание их решений.

Примечания

1 Разрешение инцидентов и проблем осуществляется посредством процессов обеспечения качества и оценки и контроля проекта. Изменения требований, архитектуры, дизайна или элементов программной системы выполняются с использованием других технических процессов.

2 Если во время работы произошел инцидент, оператор записывает инцидент (или получает автоматическое уведомление) и выполняет действия, предписанные в утвержденных рабочих процедурах, для восстановления нормальной работы. Некоторые процедуры позволяют предоставить временное решение, пока не будет проведен анализ первопричин.

3 Во время работы программного обеспечения условия, при которых возникла проблема, документируются в соответствии с поддержанием или восстановлением эксплуатационной доступности, так что, если возможно, проблема может быть продублирована в тестовой среде и выявлена основная причина. Решение проблемы включает определение серьезности и воздействия проблемы, а также определение того, следует ли исправить или принять на какое-то время проблему как известную ошибку.

3) Поддержка отслеживаемости операционных услуг и элементов конфигурации.

Примечание – Поддерживается двунаправленная прослеживаемость между эксплуатационными услугами и потребностями бизнеса или миссии, операционной концепцией, концепцией операций и требованиями заинтересованных сторон. Элементы оперативной конфигурации можно проследить до выпущенных версий и проверить через PCA или FCA.

4) Предоставление ключевых аспектов и элементов информации, выбранных для базовых показателей.

Примечание – Процесс определяет кандидатов для базового уровня, а процесс управления информацией управляет информационными элементами, такими как отчеты о производительности эксплуатационных услуг. Ключевые аспекты (информационные элементы) для Операций перечислены в Приложении B.

d) Поддержка клиента. Деятельность состоит из следующих задач:

1) Предоставление помощи и консультации клиентам и пользователям для разрешения жалоб, инцидентов, проблем и запросов на обслуживание.

Примечания

1 Помощь и консультации включают предоставление или рекомендацию ресурсов для обучения, документации, устранения уязвимостей, действий по борьбе с контрафактом и других услуг, поддерживающих эффективное использование системы программного обеспечения.

2 Поддержка клиентов может включать в себя общение с потребителями услуг, пользователями и другими заинтересованными сторонами для получения запросов на обслуживание и изменений, разрешения жалоб и предоставления информации о разрешении инцидентов и проблем.

2) Запись и отслеживание запросов и последующих действий по поддержке.

3) Определение степени, в которой поставленная программная система или услуги удовлетворяют потребности клиентов и пользователей.

Примечание – Результаты анализируются, и определяются необходимые действия по восстановлению или изменению программной системы или услуг, чтобы обеспечить постоянное удовлетворение потребностей клиентов и удобство использования программной системы. По возможности выгода от таких действий согласовывается с заинтересованными сторонами или их представителями. Данные об удовлетворенности клиентов также служат входными данными для процесса управления качеством.

6.4.13 Процесс обслуживания

6.4.13.1 Цель

Целью процесса сопровождения является поддержание способности системы предоставлять услуги.

Процесс отслеживает способность системы предоставлять услуги, регистрирует инциденты для анализа, предпринимает корректирующие, адаптивные, улучшающие и предупреждающие действия и подтверждает восстановленные возможности.

Для программных систем процесс обслуживания вносит исправления, изменения и улучшения в развернутые программные системы и элементы. Подход к обслуживанию программных систем отличается для систем, которые находятся в свободном доступе, широко распространяются на коммерческой основе или работают в небольшом количестве контролируемых сред.

Потребность в обслуживании программной системы может возникать по множеству причин, помимо скрытых системных дефектов, таких как изменения в сопряженных системах или инфраструктуре, развивающиеся угрозы безопасности и техническое устаревание элементов системы и вспомогательных систем на протяжении всего жизненного цикла системы. Расширение возможностей, обновление до середины срока эксплуатации или эволюция унаследованных систем становится новым проектом разработки программной системы, в котором будет применяться набор процессов в рамках соответствующего жизненного цикла. Если это так, то процесс управления портфелем является отправной точкой для начала работ. В других случаях обслуживание программной системы выполняется как непрерывная серия приоритетных рабочих элементов, возможно, на основе уровня усилий. Сопровождение элементов программной системы может включать аппаратное и программное обеспечение, а также сервисы, такие как коммуникационные или веб-сервисы. Сопровождение тесно связано с процессом управления конфигурацией и управлением активами программного обеспечения и выполняется параллельно с другими техническими процессами.

Примечание – [ISO/IEC/IEEE 14764:2006, ISO/IEC 16350] предоставляют дополнительную информацию. В области знаний [SWEBOK] рассматриваются основы сопровождения программного обеспечения, ключевые вопросы, измерения, методы, процесс сопровождения и вспомогательная деятельность, а также инструменты. В руководстве рассматриваются модели, методы и меры, поддерживающие надежность программного обеспечения.

6.4.13.2 Результаты

В результате успешного внедрения процесса технического обслуживания:

a) Определены ограничения на обслуживание, которые влияют на требования к системе, архитектуру или дизайн.

b) Доступны любые вспомогательные системы или услуги, необходимые для технического обслуживания.

c) Замененные, отремонтированные или измененные элементы системы доступны.

d) Сообщается о необходимости изменений для корректирующего, совершенствующего или адаптивного технического обслуживания.

e) Определяются данные об отказах и сроках службы, включая соответствующие затраты.

6.4.13.3 Деятельность и задачи

В рамках проекта должны быть выполнены следующие мероприятия и задачи в соответствии с применимыми политиками и процедурами организации в отношении процесса технического обслуживания.

a) Подготовка к техническому обслуживанию. Деятельность состоит из следующих задач:

1) Определение стратегии технического обслуживания, включая рассмотрение следующих вопросов:

i) Установление приоритетов, типовых графиков и процедур для выполнения, проверки, распространения и установки изменений в обслуживании программного обеспечения в соответствии с требованиями эксплуатационной готовности;

ii) Установление техник и методов для осознания необходимости корректирующего, адаптивного и совершенствующего обслуживания;

iii) Периодическая оценка проектных характеристик в случае эволюции программной системы и ее архитектуры;

iv) Прогнозирование потенциального устаревания компонентов и технологий с использованием информации о технических изменениях в смежных системах;

v) Установление приоритетов и ресурсов для получения доступа к правильным версиям продукта и информации о продукте, необходимой для выполнения технического обслуживания (например, плановой или поэтапной установки, исправлений для технического обслуживания или обновления программного обеспечения);

vi) Меры по техническому обслуживанию, которые позволят получить представление об уровне производительности, эффективности и результативности, включая доступ к историческим данным о неисправностях и отказах;

vii) Согласованные права на данные и воздействие на данные в системе во время решения проблем и технического обслуживания;

viii) Подход, гарантирующий, что поддельные или несанкционированные элементы системы не будут внедрены в систему;

ix) Влияние изменения обслуживания на другие элементы программных систем в сравнении с риском сохранения обнаруженной ошибки программного обеспечения;

x) Уровень квалификации и персонала, необходимый для проведения ремонта или замены системы или программного обеспечения, исправлений, патчей, обновлений и модернизаций, с учетом правовых и нормативных требований, касающихся здоровья и безопасности, безопасности и окружающей среды.

2) Для непрограммных элементов определение логистической стратегии на протяжении всего жизненного цикла, включая вопросы приобретения и эксплуатации: количество и тип заменяемых элементов, которые будут храниться, их места и условия хранения, их ожидаемый уровень замены, а также срок их хранения и частоту обновления.

Примечание – Возможности поддержки рассматриваются на ранних этапах исследования концепции или стадии разработки. Логистика помогает гарантировать, что необходимые материалы и ресурсы в нужном количестве и качества доступны в нужном месте и в нужное время на всех этапах развертывания и поддержки.

3) Определение ограничения обслуживания, которые будут включены в требования, архитектуру или дизайн системы/программного обеспечения.

Примечание – Ограничения возникают из-за необходимости: 1) повторно использовать существующие системы поддержки и проверки; 2) повторно использовать существующие запасы сменных элементов системы и учесть ограничения на пополнение запасов; 3) проводить техническое обслуживание в определенных местах или в определенных условиях. Например, программные архитектуры и проекты, которые подчеркивают инкапсуляцию, модульность и масштабируемость, могут быть проще в обслуживании. Требования к документированию проекта и конструкции системы могут снизить усилия, необходимые для обратного проектирования систем и элементов, когда необходимо техническое обслуживание. Архитектура и дизайн системы отражают необходимость возврата, резервного копирования и восстановления данных во время решения проблемы. Функции, позволяющие сделать систему доступной для удаленной диагностики и обслуживания, могут быть включены в архитектуру и дизайн.

4) Определение сделки так, чтобы система и связанные с ней действия по техническому обслуживанию и логистике привели к решению, которое было бы доступным, работоспособным, поддерживаемым и устойчивым.

Примечание – Процессы системного анализа и управления решениями используются для выполнения оценок и торговых решений.

5) Определение и планирование необходимых вспомогательных систем или услуг для поддержки технического обслуживания.

Примечание – Сюда входит определение требований и интерфейсов для вспомогательных систем. Выбор вспомогательных систем для обслуживания часто отражает необходимость повторного использования существующей или эквивалентной инфраструктуры управления проектированием, разработкой и конфигурацией, как при первоначальном внедрении системы.

6) Получение или приобретение доступа к вспомогательным системам или услугам, которые будут использоваться.

Примечание – Процесс валидации используется для объективного подтверждения того, что разрешающая система технического обслуживания достигает своего предполагаемого использования для своих разрешающих функций.

b) Выполнение технического обслуживания. Деятельность состоит из следующих задач:

1) Изучение требований, жалоб, событий, отчетов об инцидентах и проблемах заинтересованных сторон, чтобы определить потребности в корректирующем, адаптивном, улучшенном и профилактическом обслуживании.

Примечание – Для программных систем с итеративным жизненным циклом изменяющиеся требования можно рассматривать как ресурс для адаптивных и эффективных действий по обслуживанию. Для обслуживания программного обеспечения этот процесс вносит исправления, изменения и улучшения в развернутое программное обеспечение, а также устанавливает исправления и обновления для поддержания безопасности системы.

2) Анализ влияния изменений обслуживания на структуры данных, данные и связанные с ними функции программного обеспечения, пользовательскую документацию и интерфейсы.

Примечание – Обзоры и анализы включают такие факторы, как категория технического обслуживания; размер модификации; вовлеченная стоимость; время доработать; и влияет на производительность, безопасность или защищенность.

3) При обнаружении неожиданных сбоев, которые вызывают сбой программной системы, восстановление системы до рабочего состояния.

Примечание – Восстановление полного или ухудшенного рабочего состояния может быть выполнено путем возврата, обходного пути или путем выявления и исправления причины неисправности. Если полное восстановление задерживается или невозможно, система восстанавливается до ухудшенного режима в соответствии с планом действий на случай непредвиденных обстоятельств. Если возможно, ошибка реплицируется с использованием отдельной среды, аналогичной операционной среде, и выявляются первопричины неисправности. Процесс управления конфигурацией, особенно деятельность по управлению релизами, задействуется для контроля плановых и экстренных изменений в системе.

4) Внедрение процедуры по исправлению недостатков (дефектов) и ошибок, а также по замене или обновлению элементов системы.

Примечания

1 Для исправления недостатков и ошибок используется решение проблемы, которое может быть выполнено с помощью процессов обеспечения качества и оценки и контроля проекта.

2 Как правило, регрессионное тестирование выполняется для проверки того, что изменение обслуживания не привело к возникновению других проблем, т. е. для полной и правильной реализации новых и измененных требований без влияния на производительность исходных, неизмененных требований. Процесс перехода может применяться для развертывания основных изменений обслуживания; мелкие исправления обрабатываются как часть процесса обслуживания. Действия записываются для облегчения будущего обслуживания и решения проблем, а также для логистического анализа разрушаемых элементов системы.

3 Процедуры восстановления системы и данных, а также информация об обслуживании доступны на носителях, которые можно использовать при выполнении обслуживания.

5) Выполнение профилактического обслуживания путем замены, исправления, расширения или обновления элементов программной системы, чтобы повысить производительность программной системы, которая, согласно прогнозам, достигнет неприемлемых уровней обслуживания, например, нехватки емкости из-за увеличения спроса или хранимых данных, или во избежание неприемлемых условий эксплуатации, например, работа с устаревшим программным обеспечением безопасности.

6) Определение адаптивного или идеального обслуживания.

Примечание – Действия по адаптивному и усовершенствованному обслуживанию включают изменение требований, архитектуры и дизайна системы/программного обеспечения. Новый проект может быть запущен для изменения существующей системы программного обеспечения.

c) Осуществление логистической поддержки. Деятельность состоит из следующих задач:

Примечание – Логистические действия позволяют программной системе поддерживать операционную готовность. Действия включают положения о персонале, поддержке снабжения, вспомогательном оборудовании, потребности в технических данных (пользовательская документация) и согласованные права на данные, поддержку обучения, связь, поддержку оборудования/вычислительных ресурсов и помещения.

1) Получение ресурсов для поддержки программной системы на протяжении ее жизненного цикла или проекта (логистика приобретения).

Примечание – Вопросы логистики включены в соглашение, вытекающее из процессов Соглашения. Это включает в себя выполнение анализа для выявления экономически эффективных изменений в первоначальном проекте системы для обеспечения возможности поддержки и простоты обслуживания, а также договоренностей по распространению исправлений и обновлений программного обеспечения во время использования/развертывания. Решения ограничиваются требованиями доступности и влияют на управление цепочкой поставок.

2) Контроль качества и доступности сменных элементов и вспомогательных систем, их механизмы доставки и их постоянную целостность во время хранения.

Примечание – Операционная логистика включает в себя одновременную настройку, как интересующей системы, так и вспомогательных систем на протяжении всего срока эксплуатации, чтобы обеспечить эффективное и действенное выполнение функций программного обеспечения. Также включает наличие квалифицированных ресурсов. Например, доступны надежные вспомогательные системы с возможностью чтения программного обеспечения, хранящегося на предыдущих форматах носителей, или для переноса файлов резервных копий на текущие носители и поддерживаемые в настоящее время вспомогательные системы.

3) Реализация механизмов распространения программной системы или элемента, включая упаковку, обработку, хранение и связь или транспортировку, необходимые для элементов в течение жизненного цикла.

Примечания

1 Распространение и установка программного обеспечения автоматизированы. Пакеты программного обеспечения включают условия лицензии на программное обеспечение, включая права на данные и элементы для управления программными активами. Планирование логистики для других элементов системы требуется для поддержки целей процессов интеграции и перехода.

2 Необходимо учитывать хранение запасных элементов или резервных копий программного обеспечения на месте или в дополнительных местах, чтобы поддерживать возможности программной системы, по мере необходимости (возможно, на пониженном уровне для действий в непредвиденных обстоятельствах).

4) Подтверждение, что логистические действия по выполнению требований к поддержке программной системы или элемента или по достижению эксплуатационной готовности, запланированы и реализованы.

Примечание – Логистические действия могут включать в себя укомплектование персоналом, поддержку поставок, вспомогательное оборудование, потребности в технических данных (пользовательская документация, инструкции, списки), поддержку обучения, связь, поддержку оборудования / вычислительных ресурсов и помещений.

b) Управление результатами технического обслуживания и логистики. Эта деятельность состоит из следующих задач:

1) Запись инцидентов и проблем, включая их решения, а также важных результаты технического обслуживания и логистики.

Примечание – Сюда входят ошибки из-за стратегии обслуживания, систем поддержки обслуживания, выполнения обслуживания и логистики или неправильного определения системы. Процессы оценки и контроля проекта и обеспечения качества используются для выполнения идентификации и решения проблем технического обслуживания, например, анализа данных для определения первопричины, включения корректирующих или улучшающих действий и записи извлеченных уроков. Эта деятельность может включать изменения в логистике или процедурах распространения программного обеспечения. Изменения требований к программной системе, архитектуры или дизайна выполняются в рамках других технических процессов.

2) Выявление и запись тенденций инцидентов, проблем, а также действий по техническому обслуживанию и логистике.

Примечания

1 Данные о тенденциях и отчеты о решении проблем используются для информирования обслуживающего персонала, клиентов и других заинтересованных сторон и проектов, которые создают или используют аналогичные системные объекты.

2 Отчеты об инцидентах и проблемах, включая предпринятые действия, отслеживаются посредством действий по управлению инцидентами и процессами в рамках процесса обеспечения качества.

3) Поддержка отслеживаемости обслуживаемых элементов системы.

Примечание – Поддерживается двунаправленная прослеживаемость между зарегистрированными действиями по обслуживанию и элементами системы программного обеспечения и ошибками жизненного цикла. Изменения в управлении программными активами, такие как передача лицензий на программное обеспечение заменяющим системам, регистрируются.

4) Предоставление ключевых аспектов и элементов информации, выбранных для базовых показателей.

Примечание – Процесс управления конфигурацией используется для создания и поддержки элементов конфигурации и базовых показателей, а также для отслеживания лицензий и прав на данные. Процесс определяет кандидатов для базового уровня, а процесс управления информацией контролирует информационные элементы, такие как процедуры обслуживания.

5) Отслеживание и измерение степени удовлетворенности клиентов системой и техническим обслуживанием.

Примечание – [ISO 10004: 2012] содержит рекомендации по мониторингу и измерению удовлетворенности потребителей. Собранные данные об удовлетворенности клиентов используются в процессе управления качеством.

6.4.14 Процесс утилизации

6.4.14.1 Цель

Целью процесса утилизации является прекращение существования элемента системы или системы для указанного предполагаемого использования, надлежащая обработка замененных или выведенных из эксплуатации элементов и надлежащее внимание к выявленным критическим потребностям в утилизации (например, в соответствии с соглашением, в соответствии с политикой организации, или по экологическим, юридическим аспектам, вопросам безопасности).

Процесс деактивирует, разбирает и исключает систему или любой из ее элементов для конкретного использования. Он устраняет любые отходы, доводит их до окончательного состояния и возвращает окружающую среду в исходное или приемлемое состояние. Отходы могут находиться в процессе производства на любом этапе жизненного цикла, например, отходы во время производства. Этот процесс уничтожает, хранит или утилизирует элементы системы и отходы экологически безопасным образом в соответствии с законодательством, соглашениями, организационными ограничениями и требованиями заинтересованных сторон. Утилизация включает предотвращение возврата просроченных, одноразовых или неадекватных элементов в цепочку поставок. При необходимости он ведет записи, чтобы можно было контролировать здоровье операторов и пользователей, а также безопасность окружающей среды.

Утилизация программных систем включает в себя прекращение обслуживания и утилизацию программных элементов, хранимых данных, носителей и микропрограмм, информационных элементов и связанных с ними аппаратных элементов, которые не будут повторно использоваться или переходить в другую систему. Процесс утилизации предназначен для применения на любом этапе жизненного цикла программных систем. Для программного обеспечения процесс утилизации применяется на протяжении всего жизненного цикла к ресурсному коду или исполняемым копиям программного обеспечения, персонально идентифицируемым или контролируемым данным, используемым в программной системе, и связанным с ними информационным элементам, сохраняемым под централизованным контролем конфигурации или распределяемым для использования, например, утилизация прототипов на ранних стадиях жизненного цикла и вывод из эксплуатации элементов, замененных в результате модификаций на стадиях использования/развертывания и поддержки. Когда интересующая система модифицируется для модернизации технологии или возможностей, деактивируются и удаляются только затронутые элементы.

Примечание – Процесс анализа бизнеса или миссии и процесс управления решениями применяются для решения влияния на заинтересованные стороны утилизации системы и потенциальных новых возможностей системы.

6.4.14.2 Результаты

В результате успешной реализации процесса утилизации:

a) Ограничения по удалению предоставляются как входные данные для требований, архитектуры, дизайна и реализации.

b) Доступны любые вспомогательные системы или услуги, необходимые для утилизации.

c) Элементы системы или отходы уничтожаются, хранятся, утилизируются или перерабатываются в соответствии с требованиями, например требованиями безопасности.

d) Среда возвращается в исходное или согласованное состояние.

e) Доступны записи действий по утилизации и анализ.

6.4.14.3 Деятельность и задачи

В рамках проекта должны быть реализованы следующие виды деятельности и задачи в соответствии с применимыми политиками и процедурами организации в отношении процесса утилизации.

a) Подготовка к утилизации. Эта деятельность состоит из следующих задач:

1) Определение стратегии утилизации для системы программного обеспечения, чтобы включить каждый элемент системы и определить и удовлетворить критические потребности в утилизации, включая следующие соображения:

i) Окончательное прекращение функций системы и предоставление услуг, например, физическое уничтожение устройств хранения данных или передача элементов программной системы для будущего повторного использования в модифицированной или адаптированной форме;

ii) Идентификация прав собственности и ответственности за сохранение или уничтожение данных и интеллектуальной собственности в программной системе;

iii) Преобразование продукта или сохранение в социально и физически приемлемом состоянии, что позволяет избежать последующего неблагоприятного воздействия на заинтересованные стороны, общество и окружающую среду;

iv) Проблемы здоровья, безопасности, защиты и конфиденциальности, применимые к действиям по утилизации и к долгосрочному состоянию получаемых в результате физических материалов и информации;

v) Уведомление соответствующих заинтересованных сторон о значительных мероприятиях по выбытию, например, выводе из эксплуатации или замене системы, программных продуктов или услуг, графике вывода из эксплуатации или вариантах замены;

vi) Идентификация графиков, действий, ответственности и ресурсов для деятельности по утилизации.

2) Определение ограничений на утилизацию для требований к системе/программному обеспечению, архитектуры и характеристик дизайна, или методов реализации.

Примечание – Включает доступ и доступность архивов или мест долгосрочного хранения, а также доступные квалифицированные ресурсы для деактивации системы и связи с заинтересованными сторонами и партнерами по интерфейсу.

3) Определение и планирование необходимых вспомогательных систем или услуг, необходимых для поддержки утилизации.

Примечание – Включает определение требований и интерфейсов для вспомогательных систем.

1) Получение или приобретение доступа к вспомогательным системам или услугам, которые будут использоваться.

Примечание – Процесс валидации используется для объективного подтверждения того, что разрешающая система утилизации достигает своего предполагаемого использования для своих разрешающих функций.

2) Указание средств защиты, мест хранения, критериев проверки и сроков хранения, если программная система или данные должны храниться, в соответствии с соображениями безопасности и окружающей среды.

3) Определение превентивных методов, чтобы исключить повторное попадание в цепочку поставок утилизированных элементов и материалов, которые не должны быть перепрофилированы, утилизированы или повторно использованы.

b) Выполнение утилизации. Деятельность состоит из следующих задач:

1) Деактивация программной системы или элемента, чтобы подготовить их к удалению.

Примечание – Интерфейсы к другим системам рассматриваются в соответствии со специальными процедурами или инструкциями, а также соответствующими ограничениями по охране здоровья, безопасности и конфиденциальности.

2) Исключение программной системы, ее элементов, данных и материалов, не подлежащие повторному использованию, из использования или производства для соответствующего удаления и действий.

Примечание – Утилизация включает повторное использование, переработку, восстановление, капитальный ремонт или уничтожение. Распоряжение и последующие действия выполняются в соответствии с соответствующими стандартами, директивами и законами в области безопасности, защиты, конфиденциальности и охраны окружающей среды. Элементы системы программного обеспечения, у которых есть оставшийся срок полезного использования, либо в их текущем состоянии, либо после модификации, передаются другим интересующим системам или организациям. При необходимости рассмотрите возможность восстановления элементов системы, чтобы продлить их срок службы.

3) Отзыв затронутого операционного персонала из системы программного обеспечения или элемента системы и запись соответствующих операционных знаний.

Примечание – Перераспределение, передислокация или увольнение операторов. Это осуществляется в соответствии с соответствующими стандартами, директивами и законами в области безопасности, защиты, конфиденциальности и охраны окружающей среды. Действуйте, так, чтобы защитить и закрепить знания и навыки оператора, используя процесс управления знаниями.

4) Повторное использование, переработка, восстановление, капитальный ремонт, архивирование или уничтожение определенных элементов системы программного обеспечения.

Примечание – Обращаться с элементами системы и их частями, которые не предназначены для повторного использования, таким образом, чтобы гарантировать, что они не вернутся в цепочку поставок.

5) При необходимости проведение разрушения элементов системы, чтобы уменьшить объем обработки отходов или облегчить обращение с ними.

Примечание – Когда элемент не подлежит ремонту или переработке, необходимо предотвратить возвращение элементов в цепочку поставок, например, полное удаление всего программного обеспечения со всех системных носителей и удаление лицензионных ключей, данных и интерфейсов. Деятельность включает получение услуг по уничтожению для плавления, раздавливания, сжигания, сноса или уничтожения системы или ее элементов по мере необходимости.

c) Завершение утилизации. Деятельность состоит из следующих задач:

1) Подтверждение того, что вредные для здоровья, безопасности и защищенности условия окружающей среды после утилизации были выявлены и устранены.

2) Возврат среды в исходное состояние или в состояние, указанное в соглашении.

3) Архивация информации, собранной в течение всего жизненного цикла продукта, чтобы проводить аудиты и проверки в случае долгосрочных опасностей для здоровья, безопасности и окружающей среды, а также чтобы будущие создатели программных систем и пользователи могли создавать базу знаний на основе своего опыта.

# Приложение А

# (обязательное)

# Процесс адаптации

A.1 Введение

В приложении представлены требования к адаптации этого документа.

Примечания

1 Для соответствия этому документу адоптация не является требованием. Фактически, адоптация не допускается, если претензия «полное соответствие». Если заявлено об «индивидуальном соответствии», то этот процесс применяется для выполнения адаптации.

2 Дополнительное руководство по адаптации можно найти в [ISO/IEC/IEEE 24748 (все части)] по применению процессов жизненного цикла.

A.2 Процесс адаптации

A.2.1 Цель

Целью процесса адаптации является адаптация процессов данного документа к конкретным обстоятельствам или факторам:

a) заключить договор с организацией, использующей этот документ;

b) влиять на проект, который требуется для выполнения соглашения, в котором есть ссылка на этот документ;

c) отражать потребности организации в поставках продуктов или услуг.

A.2.2 Результаты

В результате успешной реализации процесса адаптации:

а) Изменены или определены новые процессы жизненного цикла для достижения целей и результатов модели жизненного цикла.

A.2.3 Деятельность и задачи

Если этот документ адаптирован, то организация или проект должны реализовать следующие задачи в соответствии с применимыми политиками и процедурами в отношении процесса адаптации, как требуется.

a) Определение и запись обстоятельства, влияющие на процесс адаптации. Влияния включают, но не ограничиваются:

1) стабильность и разнообразие операционных сред;

2) риски, коммерческие или производственные, в интересах заинтересованных сторон;

3) новизна, размер и сложность;

4) дата начала и продолжительность использования;

5) вопросы целостности, такие как безопасность, конфиденциальность, удобство использования, доступность;

6) новые технологические возможности;

7) профиль имеющихся бюджетных и организационных ресурсов;

8) доступность услуг вспомогательных систем;

9) роли, обязанности, ответственность и полномочия в общем жизненном цикле системы;

10) необходимость соответствовать другим стандартам.

b) В случае свойств, критичных для системы, необходимо должным образом учитывать структуры жизненного цикла, рекомендованные или предписанные стандартами, относящимися к размеру критичности.

c) Получение информации от сторон, которых затрагивают решения о процессе адаптации. Это включает, но не ограничивается:

1) заинтересованные стороны системы;

2) заинтересованные стороны соглашения, заключенного организацией;

3) вспомогательные организационные функции.

d) Принятие индивидуальных решений в соответствии с процессом управления решениями для достижения целей и результатов выбранной модели жизненного цикла.

Примечания

1 Организации устанавливают стандартные модели жизненного цикла как часть процесса управления моделями жизненного цикла. Иногда организации целесообразно адаптировать процессы, описанные в этом документе, для достижения целей и результатов этапов модели жизненного цикла, которая должна быть установлена.

2 Проекты выбирают организационно установленную модель жизненного цикла проекта как часть процесса планирования проекта. Иногда целесообразно адаптировать организационно принятые процессы для достижения целей и результатов этапов выбранной модели жизненного цикла.

3 В случаях, когда в проектах непосредственно применяется этот документ, иногда целесообразно адаптировать процессы этого документа для достижения целей и результатов стадий подходящей модели жизненного цикла.

e) Выбор процессов жизненного цикла, требующих адаптации, и удаление выбранных результатов, действий или задач.

Примечания

1 Независимо от адаптации, организациям и проектам всегда разрешается внедрять процессы, которые позволяют достичь дополнительных результатов или реализовать дополнительные действия и задачи, помимо тех, которые требуются для соответствия этому документу.

2 Организация или проект иногда сталкиваются с ситуацией, когда возникает желание изменить положение этого документа. Модификации следует избегать из-за непредвиденных последствий для других процессов, результатов, действий или задач. При необходимости модификация выполняется путем удаления положения (делая соответствующее заявление об индивидуальном соответствии) и, при тщательном рассмотрении последствий, внедрения процесса, который обеспечивает дополнительные результаты или выполняет дополнительные действия и задачи, выходящие за рамки адаптированного стандарта.

# Приложение В

# (рекомендуемое)

# Примеры элементов информации о процессе

В таблице B.1 представлен возможный набор рабочих продуктов, включая ошибки, записи, информационные элементы и хранилища данных, связанные с каждым процессом. Этот список не является всеобъемлющим: для каждого процесса организация может решить разработать политику, план, процедуры, отчеты и записи, чтобы продемонстрировать результаты или выполнить действия и задачи. Там, где менее интенсивная документация считается достаточной, информационные элементы могут быть объединены. Также политика и процедуры организации могут быть применены или адаптированы для каждого процесса и проекта. Показаны типичные названия элементов, включая распространенные примеры альтернативных названий в круглых скобках.

Примечание – В[ISO/IEC/IEEE 15289] приведено руководство по содержанию и управлению информационными элементами.

Артефакты, записи, хранилища записей и информационные элементы обычно инициируются в одном процессе, а пересматриваются, улучшаются или завершаются в других процессах. Для удобства они перечислены в этой таблице один раз, в процессе, где они обычно инициируются. Когда ошибки и информация программной системы преобразуются или дорабатываются другим процессом, прослеживаемость сохраняется и может быть составлено отображение прослеживаемости. Например, прослеживаемость может поддерживаться между организационными и проектными процессами, а также между требованиями, элементами архитектуры и дизайна, а также верификационными случаями.

**Таблица B.1 — Примеры информационных элементов по процессам**

# Приложение С

# (рекомендуемое)

# Эталонная модель процесса для целей оценки

C.1 Введение

Некоторые пользователи документа могут оценить внедренные процессы в соответствии со стандартами серии [ISO/IEC 33000] для оценки процессов. В приложении представлена эталонная модель процесса (PRM), подходящая для использования вместе с этими стандартами.

PRM состоит из процессов в основной части этого документа, включая название, формулировку цели и формулировку результатов для каждого процесса. В подразделе C.3 определены процессы в эталонной модели процесса и разделы, в которых они определены.

C.2 Соответствие ISO/IEC 33004

C.2.1 Общая часть

Подпункт 5.3 [ISO/IEC 33004] устанавливает требования к эталонным моделям процессов, пригодным для оценки в соответствии с этим стандартом. В следующих разделах цитируются требования [ISO/IEC 33004] к эталонным моделям процессов и описывается, как им соответствует этот документ.

C.2.2 Требования к эталонным моделям процессов

Эталонная модель процесса должна содержать [ISO/IEC 33020, 5.3.1]:

a) Объявление домена эталонной модели процесса. Это предусмотрено в разделе 1.

b) Описание взаимосвязи между эталонной моделью процесса и предполагаемым контекстом ее использования. Это предусмотрено пунктом 5.

c) Описания процессов в рамках эталонной модели процесса, отвечающие требованиям [ISO/IEC 33004, 5.4], что предусмотрено в разделе 6 описания каждого процесса.

d) Описание взаимосвязи между процессами, определенными в эталонной модели процесса. Это предусмотрено в 5.6.

Эталонная модель процесса должна документировать сообщество интересов модели и действия, предпринятые для согласования в рамках этого сообщества интересов [ISO/IEC 33020, 5.3.2]:

a) соответствующая общность интересов должна быть охарактеризована или указана в соответствии с [ISO/IEC/IEEE 15288, ISO/IEC/IEEE 12207].

b) степень достижения консенсуса должна быть задокументирована; [ISO/IEC/IEEE 15288, ISO /IEC/IEEE 12207] являются международными стандартами, удовлетворяющими согласованным требованиям [ISO/IEC, IEEE].

c) если не предпринимаются действия для согласования, заявление об этом должно быть задокументировано.

Процессы, определенные в эталонной модели процесса, должны иметь уникальные описания и идентификацию в соответствии с [ISO/IEC 33020, 5.3.3]. Идентификация обеспечивается уникальными именами и нумерацией пунктов этого приложения.

C.2.3 Описания процессов

Основными элементами эталонной модели процесса являются описания процессов в рамках модели.

Описания процессов в эталонной модели процесса включают заявление о цели, которая описывает на высоком уровне общие цели выполнения процесса вместе с набором результатов, которые демонстрируют успешное достижение цели процесса.

Описание процесса должно отвечать следующим требованиям:

a) процесс должен быть описан с точки зрения его цели и результатов;

b) набор результатов процесса должен быть необходим и достаточен для достижения цели процесса;

c) описание процессов не должно содержать или подразумевать аспекты характеристики качества процесса, выходящие за рамки базового уровня любой соответствующей структуры измерения процесса, соответствующей [ИСО/МЭК 33003].

Результат процесса описывает одно из следующего [ISO/IEC 30004: 2015, 5.4]:

a) возникновение ошибки;

b) существенное изменение состояния;

c) соответствие заданным ограничениям, например требованиям, целям и т. д.

Этим требованиям соответствуют описания процессов в разделе 6 настоящего документа. Некоторые результаты могут быть интерпретированы как способствующие повышению уровней возможностей выше уровня 1 (базового) в некоторых соответствующих системах измерения процессов. Однако согласованная реализация соответствующих процессов не требует достижения этих более высоких уровней возможностей.

C.3. Эталонная модель процесса

Эталонная модель процесса (PRM) состоит из заявления о цели и результатах каждого из процессов, включенных в раздел 6 настоящего документа. PRM для жизненного цикла программного обеспечения состоит из набора процессов на рисунке 4, показанных ранее в 5.6.1 этого документа.

# Приложение D

# (рекомендуемое)

# Интеграция процессов и конструкции процессов

D.1 Введение

Проект гармонизации в рамках [ISO/IEC JTC 1/SC 7] – параллельный, скоординированный пересмотр [ISO/IEC/IEEE 15288, ISO / IEC / IEEE 12207], а также разработка руководства в [ISO/IEC/IEEE 24748 (все части)], который содержит руководящие принципы для обоих этих документов – первый большой шаг к интегрированному набору стандартов, описывающих жизненные циклы системы и программного обеспечения. Концепции постоянного улучшения процессов и оценки возможностей в настоящее время хорошо известны и признаны, а также стандартизированы в серии стандартов [ISO/IEC 33000] (заменяющих серию [ISO/IEC 15504]). Эталонные модели процессов в Приложении C к [ISO/IEC/IEEE 15288: 2015] и в этом документе предназначены для использования вместе с [ISO/IEC 33020: 2015] для оценки возможностей процессов жизненного цикла. Определение возможностей процессов требует, чтобы описания процессов включали четкое изложение цели процесса и описание ожидаемых результатов. Последовательному внедрению процессов способствует определение действий, задач и рекомендаций. Таким образом, процессы жизненного цикла в обоих стандартах жизненного цикла приняли общие конструкции процессов, как определено в D.2 «Конструкции процессов и их использование» и согласуются с руководством по определению процессов, содержащимся в [ISO/IEC/IEEE TR 24774].

D.2 Конструкции процессов и их использование

Описания процессов в этом документе следуют четко определенным правилам. Во-первых, они были логически сгруппированы. Эти группировки продиктованы:

a) Логическими отношениями между процессами;

b) Ответственностью за выполнение процессов.

В этом документе действия, которые могут выполняться в течение жизненного цикла системы, сгруппированы в четыре группы процессов. Описание этих групп верхнего уровня можно найти в 5.6. Каждый процесс жизненного цикла в этих группах описывается с точки зрения его цели и желаемых результатов, а также перечисляются действия и задачи, которые необходимо выполнить для достижения этих результатов.

c) Процессами согласования – два процесса (6.1);

d) Организационными процессами поддержки проекта - шесть процессов (6.2);

e) Процессами технического менеджмента - восемь процессов (6.3);

f) Техническими процессами - четырнадцать процессов (6,455).

Последовательное применение правил описания процессов позволяет использовать нормализованную нумерацию пунктов. В данном документе подпункт, пронумерованный как 6.x, обозначает группу процессов, а 6.x.y - процесс в этой группе. Подпункты под номером 6.x.y.1 описывают цель процесса, подпункты под номером 6.x.y.2 описывают результаты процесса, а подпункты под номером 6.x.y.3 описывают деятельность и задачи процесса.

На рисунке D.1 представлены конструкции процесса, используемые в этом документе и в [ISO/IEC/IEEE 15288: 2015]. Это Процесс, Деятельность, Задача и Примечание.

Процесс требует названия, цели и хотя бы одного результата. У каждого процесса есть как минимум одно действие. Набор процессов с их заявлением о целях и результатах составляет эталонную модель процесса (PRM).

Деятельность представляет собой конструкцию связанных задач для получения результатов. Действия предоставляют средства для организации связанных задач в рамках процесса, чтобы улучшить понимание и коммуникацию процесса. Если деятельность достаточно связна, ее можно преобразовать в процесс более низкого уровня, сформулировав цель и набор результатов.

Задача – подробное положение о реализации процесса. Это может быть представлено как требование («должен»), рекомендация («должен») или разрешение («может»).

Примечания

1 Используются для объяснения намерений или механизмов процесса, деятельности или задачи.

2 обеспечивают понимание потенциальной реализации или областей применения, таких как списки, примеры и другие соображения.



**Рисунок D.1 - Процессные конструкции [ISO/IEC/IEEE 12207:2017, ISO/IEC/IEEE 15288:2015]**

# Приложение Е

# (рекомендуемое)

# Представления процесса

E.1 Введение

Представление процесса позволяет видеть собранный в одном месте набор действий процесса, которые прямо и лаконично решают проблему. Для таких интересов может быть разработано представление процесса, чтобы организовать процессы, действия и задачи, выбранные из [ISO/IEC/IEEE 15288 или ISO/IEC/IEEE 12207], чтобы сосредоточить внимание на их конкретной проблеме таким образом, чтобы охватить все или части жизненного цикла. В этом приложении представлены образцы точек зрения на процесс, которые можно использовать для определения представлений процесса в этих случаях.

E.2. Концепция представления процесса

Могут быть случаи, когда требуется единый фокус для действий и задач, которые выбираются из разрозненных процессов, чтобы обеспечить видимость важной концепции или потока, который пронизывает процессы, используемые на протяжении жизненного цикла. Требуется посоветовать пользователям этого документа, как идентифицировать и определять эти действия для их использования, даже если они не могут найти единственный процесс, который решает их конкретную проблему.

Для этого была сформулирована концепция представления процесса. Как и процесс, описание представления процесса включает в себя изложение цели и результатов. В отличие от процесса, описание представления процесса не включает действия и задачи. Вместо этого описание включает руководство, объясняющее, как можно достичь результатов, используя действия и задачи различных процессов в [ISO/IEC/IEEE 15288 и ISO/IEC/IEEE 12207]. Представления процесса могут быть построены с использованием шаблона точки зрения процесса, найденного в E.3.

E.3 Точка обзора процесса

Представление процесса соответствует точке обзора процесса. Точка обзора процесса, представленная здесь, может использоваться для создания представлений процесса.

1) Точка обзора процесса определяется:

1) его заинтересованными сторонами: пользователями этого документа;

2) проблемами, которые он отражает: процессами, необходимыми для отражения конкретного инженерного интереса.

2) Содержимое результирующих представлений процесса должно включать:

1) имя представления процесса;

2) цель просмотра процесса;

3) результаты просмотра процесса;

4) идентификацию и описание процессов, действий и задач, реализующих представление процесса, а также ссылки на ресурсы для этих процессов, действий и задач в других стандартах.

Примечание – Требования к документированию точек зрения можно найти в [ISO/IEC/IEEE 42010: 2011, 5.4]. Это описание в соответствии с этими требованиями.

E.4 Вид процесса для специализированного инжиниринга

В данном разделе приведен пример применения точки обзора процесса для создания представления процесса для специального инжиниринга, призванный проиллюстрировать, как проект может объединить процессы, действия и задачи данного документа для обеспечения целенаправленного внимания к достижению характеристик продукции, которые были выбраны как представляющие особый интерес.

В данном примере рассматривается группа интересов, обычно называемая специальным проектированием, которое включает, но не ограничивается такими областями, как доступность, ремонтопригодность, надежность, безопасность, защищенность, человеческие факторы и удобство использования. В данном документе эти требования к «возможностям» называются "критическими характеристиками качества". Эти характеристики определяют, насколько хорошо продукт соответствует заданным требованиям в конкретной области, выбранной для фокусировки. В E.6 приведен процесс, относящейся к (обеспечению информационной безопасности для программных систем).

Примечания

1 Обобщенный пример представления процесса, который охватывает широкий набор функциональных и нефункциональных характеристик, относящихся к специальной инженерии. Он обеспечивает широкий обзор процессов. Если конкретная критическая характеристика качества имеет высокий приоритет по сравнению с другими характеристиками, для этой характеристики можно создать конкретное представление процесса, включая более подробную информацию и требования.

2 ISO/IEC 25030 можно использовать при определении требований к качеству программного продукта.

3 Справочник по системному проектированию INCOSE содержит описания и подробные сведения о многих областях специальной инженерии и связанных с ними критических характеристиках качества.

Название: Обзор процесса специальной инженерии

Цель: Цель обзора процесса специального проектирования – предоставить объективные свидетельства того, что система достигает удовлетворительных уровней определенных критических характеристик качества, отобранных для особого внимания.

Результаты:

a) Особое внимание уделяется важнейшим качественным характеристикам продукта.

b) Определены требования к достижению критических характеристик качества.

c) Выбираются меры по требованиям, которые соотносятся с желаемыми критическими характеристиками качества.

d) Определяются и реализуются подходы к достижению желаемых критических качественных характеристик.

e) Степень выполнения требований постоянно контролируется.

f) Установлен и достигнут удовлетворительный уровень критических качественных характеристик.

Результаты допускают возможность того, что желаемые критические характеристики качества не могут быть непосредственно измерены, а вместо этого могут быть аргументированы и выведены на основе других характеристик продукта или процесса, которые могут быть измерены. Измерения могут использоваться для подтверждения соответствия установленным стандартам. Покупатель и поставщик приходят к соглашению о конкретных стандартах, которые будут использоваться для этой проверки соответствия.

Специальные инженерные процессы, мероприятия и задачи

Представление процесса может быть реализовано с использованием следующих процессов, действий и задач из этого документа.

a) Процесс оценки и контроля проекта (6.3.2) предусматривает мониторинг степени достижения требований и критических качественных характеристик, а также сообщение результатов заинтересованным сторонам и менеджерам. Соответствующие действия и задачи включают б) 6), 7), 9) и 10).

b) Процесс управления решениями (6.3.3) обеспечивает оценку альтернативных требований, характеристик архитектуры и проектных характеристик по критериям принятия решения, включая критические характеристики качества. Результаты этих сравнений ранжируются с помощью подходящей модели выбора, а затем используются для выбора оптимального решения. Соответствующие действия и задачи включают b) (все задачи); и в) 1).

c) Процесс управления рисками (6.3.4) в целом обеспечивает идентификацию, оценку и обработку рисков системы, в том числе связанных с соблюдением критических характеристик качества.

d) Процесс управления информацией (6.3.6), в полном объеме, предусматривает спецификацию, разработку и поддержание информационных элементов для документирования и передачи информации о степени достижения. Информационные элементы, используемые для целей критических характеристик качества, иногда имеют специализированный характер. Ресурсы для описания этих информационных элементов включают отраслевые ассоциации, регулирующие органы и специальные стандарты

e) Процесс измерения (6.3.7) в целом предусматривает определение подхода, который связывает меры с необходимыми критическими характеристиками качества.

f) Процесс обеспечения качества (6.3.8) направлен на выявленные ошибок (инциденты и проблемы), которые связаны с достижением критических характеристик качества.

g) Процесс анализа бизнеса и миссии (6.4.1) обеспечивает определение проблемного пространства и характеристику пространства решений, включая соответствующие факторы торгового пространства и предварительные концепции жизненного цикла. Это включает в себя развитие понимания контекста и любых ключевых параметров, таких как критические характеристики качества, например, угрозы безопасности, человеческий интерфейс, рабочие характеристики и контекст обеспечения безопасности системы. Соответствующие действия и задачи включают б) 1) и 2); в) 1); и г) 1).

h) Процесс определения потребностей и требований заинтересованных сторон (6.4.2) предусматривает выбор и определение характеристик, включая критические характеристики качества, и связанные с ними элементы информации. Действия и документация полезны для выявления, приоритезации, определения и регистрации требований для критических характеристик качества. Соответствующие действия и задачи включают а) 1) и 2); б) 2), 3) и 4); в) 1) и 2); г) все задачи; и д) 2).

i) Процесс определения требований к системе/программному обеспечению (6.4.2) предусматривает спецификацию параметров для критических характеристик качества и выбор мер для отслеживания достижения этих требований по отношению к конкретной разрабатываемой системе. Соответствующие действия и задачи включают а) 1); б) все задачи; и в) 2).

j) Процесс определения архитектуры (6.4.4) обеспечивает идентификацию проблем заинтересованных сторон с точки зрения архитектуры. Эти опасения часто трансформируются в ожидания или ограничения на этапах жизненного цикла, которые связаны с критическими характеристиками качества, такими как использование, например, доступность, безопасность, эффективность, удобство использования; поддержка, например, ремонтопригодность, управление моральным износом; эволюция системы и окружающей среды, например, адаптивность, масштабируемость, живучесть; производство, например, технологичность, тестируемость; вывод из эксплуатации, например, воздействие на окружающую среду, транспортабельность. Этот процесс дополнительно обращается к тем критическим требованиям к характеристикам качества, которые определяют архитектурные решения, включая оценку архитектуры в отношении проблем и связанных характеристик. Соответствующие действия и задачи включают а) 2) и 4); б) 1); в) 2), 3), 4) и 5); г) 1); и д) 2).

k) Процесс определения проекта (6.4.5) обеспечивает определение необходимых проектных характеристик, которые включают критические характеристики качества, такие как надежность критериев проектирования для специальных характеристик и оценка альтернативных проектов по этим критериям. Соответствующие действия и задачи включают а) 2); б) 1), 2), 3) 4) и 6); и в) 2).

l) Процесс системного анализа (6.4.6) обеспечивает уровень анализа, необходимый для понимания торгового пространства в отношении критических характеристик качества посредством проведения математического анализа, моделирования, экспериментов и других методов. Результаты анализа вводятся в сделки, совершаемые через процесс управления решениями в поддержку других технических процессов. Соответствующие действия и задачи включают: а) (все задачи); и б) (все задачи).

m) Процесс внедрения (6.4.7) предусматривает регистрацию свидетельств того, что критические требования к качеству были выполнены. Соответствующие действия и задачи включают б) 3).

n) Процесс интеграции (6.4.8) обеспечивает планирование интеграции, включая рассмотрение критических характеристик качества, а также уверенность в том, что достижение характеристик определяется и регистрируется. Соответствующие действия и задачи включают а) 1); б) 3); и в) 1).

o) Процесс верификации (6.4.9) предусматривает планирование и выполнение стратегии выполнения верификации, включая критические характеристики качества. Выбранная стратегия проверки может вводить конструктивные ограничения, влияющие на достижение характеристик. Соответствующие действия и задачи включают а) 1) и 3); б) 1), 2); и в) 1) и 2).

p) Процесс перехода (6.4.10) предусматривает установку системы в ее операционной среде. Поскольку некоторые специальные свойства предполагают компромисс между конструктивными и эксплуатационными ограничениями, часто важно уделять внимание установке. Соответствующие действия и задачи включают а) 4); и б) 4), 6) и 7).

q) Процесс валидации (6.4.11) предоставляет свидетельство того, что услуги, предоставляемые системой, удовлетворяют потребности заинтересованных сторон, включая критические характеристики качества. Соответствующие действия и задачи включают а) 1) и 3); б) 1) и 2); в) 1) и 2).

r) Процесс эксплуатации (6.4.12) предусматривает использование системы. Обеспечение надлежащего достижения критических характеристик качества включает мониторинг работы системы. Соответствующие действия и задачи включают б) 3) и 4); в) 1) и 2); и г) 1) и 2).

s) Процесс сопровождения (6.4.13) поддерживает возможности системы, особенно ее постоянную доступность для обеспечения своих функций, включая критические характеристики качества. Сюда входят анализ отказов, задачи обслуживания и задачи логистики, необходимые для обеспечения непрерывной работы системы. Соответствующие действия и задачи включают б) все задачи; в) все задачи; и г) 1) и 2).

t) Процесс удаления (6.4.14) прекращает существование системы. Неотъемлемая необходимость предвидеть утилизацию может наложить ограничения на разработку. Фактически, эти ограничения могут быть критическими характеристиками качества. Соответствующие действия и задачи включают а) 2); б) 1) и 2) и в) 3).

E.5 Представление процесса для управления интерфейсом

В разделе представлен пример применения точки обзора процесса для получения представления процесса управления интерфейсом, предназначенный для иллюстрации того, как проект может объединять процессы, действия и задачи этого документа, чтобы уделять особое внимание достижению характеристик продукта, которые были выбраны представляющие особый интерес.

В примере рассматривается конкретный экземпляр представления процесса, называемый управлением интерфейсом, который включает определение интерфейса, дизайн и управление изменениями. В этом документе задачи, составляющие управление интерфейсом, полностью содержатся в существующих процессах.

Примечание – Для программных систем интерфейсы с другими системами являются типичным способом получения входных данных или экспорта выходных данных. данные (отчеты). Интерфейсы с внешними системами и службами позволяют программной системе функционировать в своей операционной среде и с вспомогательными системами. Графический интерфейс пользователя – специальный интерфейс для взаимодействия человека с программной системой.

Название: Просмотр процесса управления интерфейсом.

Цель: Цель обзора процесса управления интерфейсами – облегчить идентификацию, определение, проектирование и управление интерфейсами программной системы.

Результаты:

a) Идентифицируются бизнес-потребности или миссии, связанные с интерфейсами.

b) Выявлены потребности заинтересованных сторон, связанные с интерфейсами

c) Определены требования к интерфейсам.

d) Идентифицируются и определяются интерфейсы между элементами системы программного обеспечения.

e) Идентифицируются и определяются интерфейсы между программной системой и внешними системами.

f) Степень реализации требований к интерфейсу постоянно контролируется.

Процессы, действия и задачи управления интерфейсом

Представление процесса может быть реализовано с использованием следующих процессов, действий и задач из этого документа.

Примечание – Справочник по системной инженерии INCOSE содержит описания и подробные сведения об управлении интерфейсом.

a) Процесс оценки и контроля проекта (6.3.2 Закладка «Ошибка!» Не определен.) Обеспечивает мониторинг степени выполнения требований, включая интерфейсы, и передачу результатов заинтересованным сторонам и лицам, принимающим решения. Соответствующие действия и задачи включают б) 6), 7), 9) и 10).

b) Процесс управления принятием решений (6.3.3) обеспечивает оценку альтернативных требований, характеристик архитектуры и проектных характеристик по критериям принятия решения, включая интерфейсы. Результаты этих сравнений ранжируются с помощью подходящей модели выбора и затем используются для принятия решения об оптимальном решении. Соответствующие действия и задачи включают б) все задачи); и в) 1).

c) Процесс управления рисками (6.3.4) в целом обеспечивает идентификацию, оценку и обработку рисков системы, в том числе связанных с интерфейсами.

d) Процесс управления конфигурацией (6.3.5) обеспечивает идентификацию интерфейсов и управление ими. Он включает в себя управление спецификациями интерфейсов и документами по управлению интерфейсами. Требования и изменения к внутреннему и внешнему интерфейсу документируются в соответствии со стратегией CM проекта, которая обычно представлена в плане CM. Соответствующие действия включают b) 1) и d) 1).

e) Процесс управления информацией (6.3.6) в целом предусматривает спецификацию, разработку и обслуживание элементов информации для документирования интерфейсов и их эксплуатационных характеристик.

f) Процесс измерения (6.3.7) в целом предусматривает определение подхода, который связывает меры с требуемыми потребностями в информации интерфейса, а затем генерирует и использует эти меры для удовлетворения выявленных потребностей в информации интерфейса.

g) Процесс обеспечения качества (6.3.8) в целом направлен на устранение выявленных ошибок (инцидентов и проблем), которые связаны с выполнением требований интерфейса.

h) Процесс анализа бизнеса и миссии (6.4.1) обеспечивает определение проблемного пространства и характеристику пространства решений, включая описание среды и контекста, а также предварительные операционные концепции, включая программные интерфейсы системы и обеспечивающие системные интерфейсы. Он часто определяет внешние системы, которые взаимодействуют с интересующей системой. Соответствующие действия и задачи включают б) 1) и 2); и в) 1).

i) Процесс определения потребностей и требований заинтересованных сторон (6.4.2) обеспечивает идентификацию заинтересованных сторон, связанных с интерфейсами, определение операционных концепций и взаимодействия системы с пользователями, существующие интерфейсы, которые должны быть переданы, и предполагаемую среду (включая другие системы). Он часто определяет внешние системы, которые взаимодействуют с интересующей системой. Соответствующие действия и задачи включают а) 1), в) 1) и 2), и г) 1) и 3).

j) Процесс определения требований к системе/программному обеспечению (6.4.3) обеспечивает определение границ системы и требований к интерфейсу. Соответствующие действия и задачи включают а) 1); б) 3) и 4); c) (все задачи) и d) 1) и 3).

k) Процесс определения архитектуры (6.4.4) обеспечивает идентификацию интерфейсов с точки зрения архитектуры по мере развития архитектурных моделей. Этот процесс дополнительно описывает и определяет интерфейсы в той степени, в которой это необходимо для описания архитектуры. Соответствующие действия и задачи включают а) 2); в) с 1) по 4); г) 2); и е) 2) и 6).

l) Процесс определения проекта (6.4.5) обеспечивает уточнение и полное определение интерфейсов и создание элементов информации для определения характеристик интерфейсов и протоколов. Соответствующие действия и задачи включают б) 2), 5) и 6); в3) и г) 2).

m) Процесс системного анализа (6.4.6) обеспечивает уровень анализа, необходимый для уравновешивания архитектуры интерфейса, проектных ограничений, требований к рабочим характеристикам интерфейса и измерений эксплуатационных характеристик посредством проведения моделирования, имитации и других методов. Результаты анализа вводятся в процесс управления решениями в поддержку других технических процессов. Соответствующие действия и задачи включают: а) (все задачи); и (б) (все задачи).

n) Процесс реализации (6.4.7) предусматривает разработку интерфейсов и запись свидетельств того, что требования к интерфейсу для реализованного элемента системы были выполнены. Соответствующие действия и задачи включают б) 1) и 3).

o) Процесс интеграции (6.4.8) предусматривает планирование интеграции, включая рассмотрение интерфейсов между элементами программной системы и внешними системами. Он также включает в себя интеграцию систем или системных элементов и интерфейсов и подтверждение интерфейсов в интегрированной системе программного обеспечения. Соответствующие действия и задачи включают а) 1), 2) и 5); б) все задачи; и в) 1).

p) Процесс верификации (6.4.9) обеспечивает свидетельство того, что услуги, предоставляемые системой, соответствуют системным требованиям, включая требования к интерфейсу. Процесс предусматривает планирование и выполнение стратегии для выполнения проверки, включая требования к интерфейсу. Выбранная стратегия проверки может вводить ограничения интерфейса, которые влияют на их реализацию. Соответствующие действия и задачи включают а) 1) и 3); б) 1), 2); и в) 1).

q) Процесс перехода (6.4.10) предусматривает планирование и миграцию программной системы или элементов и наборов данных в другую среду с использованием интерфейсов и установление использования интерфейсов в новой системе. Это включает в себя определение ограничений и проверку установки, активации и рабочего состояния интерфейсов. Соответствующие действия и задачи включают а) 1) и 3); б) 3), 4), 6) и 7).

r) Процесс валидации (6.4.11) предоставляет свидетельство того, что услуги, предоставляемые программной системой, удовлетворяют потребности заинтересованных сторон, включая требования к интерфейсу. Выбранная стратегия проверки может вводить ограничения интерфейса, которые влияют на их реализацию. Валидация включает в себя общение с представителями заинтересованных сторон взаимодействующих систем и сервисов. Соответствующие действия и задачи включают а) 4); б) 1) и 2); в) 1) и 2).

s) Процесс эксплуатации (6.4.12) предусматривает использование программного комплекса. Также могут быть ограничения на интерфейсы для операций. Подтверждение того, что требования к интерфейсу выполнены надлежащим образом, включает мониторинг работы системы, определение и выполнение корректирующих действий, когда интерфейсы не работают должным образом, а также поддержку контактов с партнерами по интерфейсу (клиентами). Соответствующие действия и задачи включают а) 1) и 2), б) 1), 3) и 4); в) 1) и 2) и г (все задачи).

t) Процесс сопровождения (6.4.13) поддерживает возможности программной системы, включая ее интерфейсы, и их постоянную доступность для обеспечения своих функций. Сюда входят задачи анализа проблем и технического обслуживания, а также задачи логистики, необходимые для обеспечения непрерывной и своевременной работы. Соответствующие действия и задачи включают а) 2); б) (все задачи); и г) 1), 2) и 3).

u) Процесс удаления (6.4.14) завершает существование системы или интерфейса. Это включает в себя действия по отключению и прекращению взаимодействия. Соответствующие действия и задачи включают а) 2) и 3); и б) 1), 2) и 6).

E.6 Представление процесса обеспечения безопасности программного обеспечения (Информационная безопасность)

В разделе представлен пример применения точки зрения процесса для получения представления процесса для обеспечения программного обеспечения, предназначенный для иллюстрации того, как проект может объединять процессы, действия и задачи этого документа, чтобы уделять особое внимание достижению выбранных характеристик обеспечения качества программного обеспечения. как представляющий особый интерес. Характеристики обеспечения программного обеспечения, степень их достижения и соответствующая информация могут подтверждать заявление о гарантии программного обеспечения, как описано в [ISO/IEC/IEEE 15026].

Пример сфокусирован на защите от преднамеренной подрывной деятельности или принудительного отказа из-за архитектуры, дизайна или реализации программного обеспечения, особенно построения кода.

Название: Просмотр процесса обеспечения безопасности ПО (гарантия безопасности программного обеспечения).

Цель: Цель обзора процесса обеспечения безопасности ПО – предоставить объективные свидетельства того, что программное обеспечение достигает удовлетворительного уровня уверенности в том, что достигается достаточная защита от преднамеренной подрывной деятельности или принудительного отказа из-за архитектуры программного обеспечения, дизайна или построения кода.

Примечание – В условия, подходящие для соответствия [ISO/IEC/IEEE 15026], гарантийные заявления относительно характеристик программного обеспечения, выбранных для особого внимания, достигаются, и предоставляется информация, показывающая выполнение этих требований.

Результаты:

a) Особое внимание уделяется характеристикам обеспечения качества программного обеспечения продукта.

b) Определены требования к достижению характеристик доверия к программному обеспечению.

c) Выбираются меры для требований, которые соотносятся с желаемыми характеристиками обеспечения программного обеспечения.

d) Определяются и реализуются подходы к достижению желаемых характеристик гарантии программного обеспечения.

e) Степень выполнения требований постоянно контролируется.

f) Удовлетворительный уровень критических характеристик гарантии программного обеспечения определен и достигнут.

Результаты допускают возможность того, что желаемые характеристики или утверждение программного обеспечения не могут быть непосредственно измерены, а вместо этого могут быть выведены на основе других характеристик продукта или процесса, которые могут быть измерены. Измерения могут использоваться для подтверждения соответствия установленным стандартам. Покупатель и поставщик договариваются о конкретных стандартах, которые будут использоваться для этой проверки соответствия.

Процессы, действия и задачи обеспечения безопасности ПО

Представление процесса может быть реализовано с использованием следующих процессов, действий и задач из этого документа.

a) Процессы Соглашения (6.1) предусматривают установление ожиданий и обязанностей, связанных с обеспечением безопасности программного обеспечения, включая юридические соглашения и лицензионные требования, защиту активов организаций, которые доступны поставщикам, возможность компрометации во время доставки, обнаружение аномалий, обнаружение о подделках в элементе приложения по прибытии, ожиданиях по устранению дефектов, управлении исправлениями, соглашениях об уровне обслуживания и мерах защиты от угроз цепочки поставок. Соответствующие действия и задачи включают в процесс приобретения: а) 1) и 2) и в) 1); и в Процессе поставки, c) 1), e) 1 и 2).

b) Процесс управления моделью жизненного цикла (6.2.1) обеспечивает создание и поддержку политик и процедур обеспечения программного обеспечения, включая жизненный цикл разработки безопасности программного обеспечения и контролируемое использование кода, переданного на аутсорсинг. Соответствующие действия и задачи включают а) 2 и 3), б) 1 и в) все задачи.

c) Процесс управления инфраструктурой (6.2.2) в целом обеспечивает безопасные среды разработки и эксплуатации, инструменты обеспечения безопасности программного обеспечения, своевременное управление исправлениями и библиотеки кода, которые должным образом обслуживаются и улучшаются на основе уроков, извлеченных из проектов, организации и отрасли. .

d) Процесс управления человеческими ресурсами (6.2.4) в целом предусматривает соответствующую проверку сотрудников и поставщиков, имеющих доступ к программному обеспечению или среде разработки программного обеспечения, и определяет специальное обучение, связанное с обеспечением безопасности программного обеспечения, на основе ролей и обязанностей на протяжении жизненного цикла.

e) Процесс управления знаниями (6.2.6) обеспечивает сбор и поддержание знаний и информирование проекта об изменениях в наборе угроз, эволюции практики обеспечения программного обеспечения, смягчении уязвимостей программного обеспечения, уроках, извлеченных из инцидентов и реагировании на инциденты, а также эволюции в инструменты тестирования программного обеспечения. Соответствующие действия и задачи включают b) (все задачи) и d) 3).

f) Процесс управления принятием решений (6.3.3) в целом оценивает альтернативные требования, характеристики архитектуры и характеристики проекта в сравнении с критериями принятия решения, включая характеристики обеспечения качества программного обеспечения. Результаты этих сравнений ранжируются и записываются вместе с моделью выбора, а затем используются для принятия решения об оптимальном решении. Заинтересованные стороны могут принимать решения на основании предоставленного обоснования.

g) Процесс управления рисками (6.3.4) в целом предусматривает оценку возможности неспособности достичь необходимой безопасности приложения, что приводит к риску для пользователей, включая партнеров по интерфейсу, или к тому, что программное обеспечение не используется по назначению. Безопасность программного обеспечения - это категория риска в каждом анализе рисков.

h) Процесс управления конфигурацией (6.3.5) обеспечивает создание и поддержание целостности всех идентифицированных выходных данных проекта или процесса, включая мониторинг активов и безопасность назначенных систем хранения, и делает их доступными для уполномоченных сторон. Сюда входят записи об изменениях, внесенных в программное обеспечение и его выпуски, а также контроль и аудит всех обращений к программным элементам, которые выполняют функции безопасности. Соответствующие действия и задачи включают a) 1), d) 1) и f) 2).

i) Процесс управления информацией (6.3.6) в целом предоставляет информацию о достижении гарантии программного обеспечения соответствующим заинтересованным сторонам, включая регулирующие органы или органы утверждения. Поддающаяся количественной оценке информация о программном обеспечении собирается для поддержки ряда аргументов, которые оправдывают заявление о программном обеспечении системы. Из-за чувствительности данных особое внимание уделяется определению подходящей аудитории для различных мер доверия. Управление информацией включает в себя защиту конфиденциальной информации в отношении обеспечения программного обеспечения.

j) Процесс измерения (6.3.7) в целом обеспечивает общую платформу для сбора информации о заявлениях, стратегиях и доказательствах в отношении программного обеспечения, которые иногда называются случаем, обеспечивающим уверенность.

k) Процесс обеспечения качества (6.3.8) оценивает процессы проекта и поставщиков на соответствие требованиям и процедурам обеспечения качества программного обеспечения. Он устраняет выявленные ошибки (инциденты и проблемы), которые связаны с достижением характеристик обеспечения качества программного обеспечения. Соответствующие действия и задачи включают c) и e) (все задачи).

l) Процесс анализа бизнеса и миссии (6.4.1) обеспечивает понимание операционной среды для разрабатываемой программной системы, а также законов, политик, рисков и ограничений, связанных с гарантиями программного обеспечения. Соответствующие действия и задачи включают b) 1) и c) 1).

m) Процесс определения потребностей и требований заинтересованных сторон (6.4.2) предусматривает выбор и определение рисков и угроз для миссий или информации. Он включает эти знания при определении требований, относящихся к обеспечению безопасности программного обеспечения (безопасность информации), включая конфиденциальность, доступность и целостность в контексте того, как программное обеспечение предназначено для работы; и с учетом сценариев неправомерного использования и злоупотреблений. Заинтересованным сторонам необходимо договориться о том, каких аспектов обеспечения программного обеспечения достаточно. Соответствующие действия и задачи включают а) 2); б) 1) и 2); и c) (все задачи).

n) Процесс определения требований к системе / программному обеспечению (6.4.3) предусматривает выбор и определение требований, связанных с обеспечением безопасности программного обеспечения (информационная безопасность), включая конфиденциальность, доступность и целостность в контексте того, как программное обеспечение предназначено для работы; и требования к целостности программного обеспечения в случае неправильного использования и злоупотребления. Соответствующие действия и задачи включают б) 3) и 4); и в) 3).

o) Процесс определения архитектуры (6.4.4) обеспечивает идентификацию проблем заинтересованных сторон с точки зрения архитектуры посредством моделирования угроз и оценки уязвимости архитектуры и дизайна продукта для потенциальных атак, чтобы получить представление о ландшафте угроз и соответствующих элементах архитектуры. Соответствующие действия и задачи включают а) 2) и 4); б) 1); c) (все задачи), d) 5) и f) 10, 20 и 5).

p) Процесс определения проекта (6.4.5) обеспечивает определение необходимых проектных характеристик, которые включают уменьшение видов атак, включая расположение компонентов; шаблоны проектирования обеспечения программного обеспечения; и избегание шаблонных ошибок. Соответствующие действия и задачи включают а) 2) и 3); б) 2), 3) и 6); в) 2) и г) 1).

q) Процесс реализации (6.4.7) предусматривает использование методов безопасного кодирования, чтобы избежать распространенных ошибок кодирования, которые приводят к использованию уязвимостей продукта, а также использование различных методов тестирования, включая проверку на компетентную аутентичность, нечеткое тестирование, тестирование статическим анализом и т. д. и динамическое тестирование для выявления и устранения слабых мест и уязвимостей программного обеспечения. Соответствующие действия и задачи включают а) 1); и б) 1), 2) 3) и 4).

r) Процесс интеграции (6.4.8) обеспечивает планирование интеграции, включая рассмотрение характеристик обеспечения программного обеспечения, а также уверенность в том, что достижение характеристик определяется и регистрируется. Внедрение стандартов интерфейсов везде, где это возможно, способствует устойчивости системы и элементов, а также возможности повторного использования элементов. Соответствующие действия и задачи включают а) 2) и 5); б) 3); и в) 1) и 2).

s) Процесс верификации (6.4.9) предусматривает планирование и выполнение стратегии для проверки того, что требования к программному обеспечению, включая характеристики доверия к программному обеспечению, были достигнуты. Выбранная стратегия проверки может включать тестирование слабых мест кода в процессе разработки или во время поддержки. Анализ угроз обеспечивает исходные данные для создания планов тестирования и кейсов. Результаты включают в себя информацию, необходимую для выполнения корректирующих действий, которые исправляют несоответствия в программном обеспечении или процессах, которые воздействуют на него, и учитывают неопределенность в действиях по верификации, такую как надежность инструмента тестирования и уровень неопределенности в результатах (т. е. ложных срабатываний и ложных отрицаний).

Дополнительные соображения включают отказоустойчивость программного обеспечения для работы. Соответствующие действия и задачи включают а) 1), 4), 5) и 6: б) (все задачи); в) 1) и 2).

t) Процесс перехода (6.4.10) предусматривает установку программной системы или элемента в другой среде. Поскольку некоторые свойства Software Assurance предполагают компромисс между конструктивными и эксплуатационными ограничениями, часто важно уделять внимание установке и пользовательской документации. Соответствующие действия и задачи включают а) 2); б) 1) и 4), 5) и 6); и в) 1) и 2).

u) Процесс валидации (6.4.11) предоставляет свидетельство того, что услуги, предоставляемые системой программного обеспечения, удовлетворяют потребности заинтересованных сторон, включая характеристики гарантии программного обеспечения. Методы проверки включают сканирование уязвимостей, оценку и проверку кода с использованием различных инструментов и методов, таких как статический анализ кода, динамический анализ кода, анализ двоичного кода, инструменты покрытия кода, стресс-тестирование и использование инструментов для сбора свидетельств изменений, вызванных удаленное техническое обслуживание. Дополнительные соображения включают устойчивость программного обеспечения к работе в случаях использования и злоупотреблений. Соответствующие действия и задачи включают: a) 1), 3 и 4), b (все задачи) и c) 2) и 3).

v) Процесс эксплуатации (6.4.12) предусматривает использование программного комплекса. Обеспечение надлежащего достижения характеристик программного обеспечения включает в себя мониторинг работы системы для предоставления ее услуг в предполагаемой среде и оказание поддержки клиентам программного продукта. Планы этого процесса учитывают достижение безопасности приложений на протяжении всего жизненного цикла системы, эксплуатационные ограничения, такие как контроль доступа, и согласованность предположений, сделанных на более ранних этапах относительно безопасности приложений с операционной средой. Процесс включает создание систем отчетности и процедур для расследования и устранения инцидентов, связанных с безопасностью приложений. Соответствующие действия и задачи включают а) 1), б) (все задачи); в) 1), 2) и 3); и г) 1).

w) Процесс сопровождения (6.4.13) поддерживает возможности программной системы, особенно ее постоянную доступность для обеспечения характеристик гарантии программного обеспечения. Сюда входят анализ отказов, задачи обслуживания и задачи логистики, необходимые для обеспечения непрерывной работы системы. Процесс сопровождения обеспечивает оценку влияния на Software Assurance изменений, внесенных в программное обеспечение во время обслуживания, и поддержание соответствующих свидетельств для Software Assurance. Он включает в себя документированный процесс установки исправлений и исправлений программного обеспечения, обнаружения, удаления или деактивации неавторизованного и вредоносного программного обеспечения, а также процесс информирования эквайера или партнера по интерфейсу с помощью механизмов уведомления и исправления. Устранение уязвимостей имеет приоритет на основе множества факторов, включая риск. При реализации исправления программного обеспечения соблюдаются документированные методы разработки и поддержки. Соответствующие действия и задачи включают а) 1); б) (все задачи); в) (все задачи); и г) 1) и 2).

x) Процесс удаления (6.4.14) прекращает существование программной системы. Неотъемлемая необходимость предвидеть удаление может накладывать ограничения на разработку и управление данными, содержащимися в системе программного обеспечения. Фактически, эти ограничения могут быть характеристиками гарантии программного обеспечения. Соответствующие действия и задачи включают а) 1), 2) и 5); б) (все задачи); и в) 1) и 3).

# Приложение F

# (рекомендуемое)

# Моделирование архитектуры программных систем

F.1 Введение

В документе деятельность по архитектуре и дизайну описывается как отдельные процессы. При развитии архитектуры программного элемента могут быть задействованы несколько итераций процессов определения архитектуры, определения дизайна и реализации, например, для определения того, будет ли реализована сущность или функция посредством интеграции существующего программного обеспечения, адаптивного повторного использования или вновь созданного программного обеспечения. В сообществах системной и программной инженерии, имеющих дело со сложными системами, за архитектурой могут следовать разные конструкции для разных систем в разных линейках продуктов. В этом случае важно выполнять эти два процесса по отдельности. Кроме того, архитектура часто делается по другим причинам, а не как непосредственная основа для проектирования, например, для стимулирования инвестиций в технологии.

Архитектура программной системы может пониматься как набор структурированных архитектурных объектов и их взаимосвязей, выбранных для достижения таких характеристик, как функциональная совместимость, масштабируемость, устойчивость к окружающей среде, инкапсуляция, доступность, надежность, эффективность выполнения или эффективность миссии (пригодность для использования). Архитектура программной системы имеет дело с отношениями между различными объектами, такими как сценарии, функции, потоки функций, интерфейсы, элементы потока ресурсов, информация или элементы данных, объекты, физические компоненты и среды, контейнеры, узлы, ссылки, ресурсы связи, ограничения, уравнения и параметрические модели.

В приложении описаны некоторые модели (типы моделей), которые используются при создании и оценке архитектуры программных систем.

Примечание – [ISO/IEC/IEEE 15288: 2015, Приложение F] описывает, как модели и представления применяются к архитектуре систем в целом, и имеет дополнительную информацию, касающуюся видов и моделирования архитектуры физических продуктов, таких как массовые модели и макеты.

F.2 Представления, модели и виды моделей, используемые в архитектуре программных систем

В процессе определения архитектуры используются различные модели программных систем, включая примеры моделей, перечисленные в следующем разделе. Типы моделей определяют языки, обозначения, соглашения, методы моделирования, аналитические методы или другие операции, которые будут использоваться с моделями такого типа. (Традиционная практика системной инженерии классифицирует некоторые из этих моделей как «логические модели» или «физические модели», но таксономическое различие не требуется при применении этого документа.) Для представления того, как архитектура системы решает проблемы заинтересованных сторон, используются различные представления. Представления состоят из моделей. Например, логическое представление программного обеспечения может представлять бизнес-процессы в его функциях; представление процесса может представлять события и преобразования, происходящие в различных состояниях программного обеспечения, и может включать вопросы параллелизма и синхронизации; структурное представление представляет различные компоненты системы, которые могут быть связаны с физическими или виртуальными элементами системы, представление информации представляет отношения между элементами данных, содержащимися в программном обеспечении и преобразованными им.

Определения терминов архитектуры и дополнительную информацию о концепциях и моделях архитектуры приведены в [ISO/IEC/IEEE 42010].

F.2.1 Функциональная модель

Функциональная модель системы – представление набора функций, которые определяют преобразования входов в выходы, выполняемые системой для достижения своей миссии или цели. Эти функции определяются ожидаемым поведением системы при использовании по назначению. Следовательно, каждая системная функция связана с взаимодействием между системой и окружающей средой. Функциональные требования, требования к производительности, нефункциональные требования и ограничения обычно анализируются для определения функций и потоков ввода-вывода. Когда функции связаны с элементами системы, в процессе определения проекта необходимо определить, достаточно ли каждый элемент системы/программного обеспечения определен, чтобы построить или купить его. Если системный элемент дополнительно разрешен для достижения этой достаточности, тогда функции, связанные с системным элементом, также будут дополнительно разрешены и должным образом связаны с подэлементами. Как правило, существует несколько способов декомпозиции функций, которые способствуют определению нескольких архитектур-кандидатов.

F.2.2 Статическая модель

Статическая модель описывает структуру программной системы. В объектно-ориентированном программировании он представлен в виде набора объектов (классов) и их отношений (наследования, ассоциации и зависимости), представленных в виде узлов и ссылок.

F.2.3 Модель данных

Модель данных (семантическая или информационная модель) представляет элементы данных, их отношения и свойства (атрибуты), которые обрабатываются программной системой. Логические модели данных используют схему для отражения структурных отношений между объектами данных, которые могут быть реализованы в базах данных. Модели данных отражают различные типы данных (текст, графика, географические данные, изображения, общие объекты) и их использование в функциях системы (частота изменения, объем данных, использование при поиске), а также логические отношения между элементами данных. Модели данных применяются при разработке интерфейсов и программных сервисов, анализе данных и составлении отчетов. Физические модели данных отражают схему хранения и поиска записей данных.

F.2.4 Поведенческая модель

Поведенческая модель (динамическая модель) – набор функций и интерфейсов (внутренних и внешних), который определяет, как система или ее элементы действуют в условиях, поддерживающих рабочие сценарии, включая последовательность выполнения, синхронизацию и параллелизм, условия для поведенческого изменение и производительности. Поведенческие модели применимы к программным системам управления. Поведенческую модель можно описать набором взаимосвязанных сценариев. Включает в себя идентификацию поведенческих элементов (например, режимов/состояний, переходов, триггерных событий и рабочих сценариев) в течение жизненного цикла.

F.2.5 Временная модель

Временная модель системы – представление, которое выражает, как время учитывается в поведении системы или ее элементов, которое представляет уровни частоты выполнения функций (например, стратегический уровень, тактический уровень, уровень оперативного мониторинга, уровень регулирования) соответствующие уровням принятия решений, которые позволяют пользователям и программной логике отслеживать и контролировать работу системы. Включает определение временных элементов (например, продолжительности, частоты, времени отклика, триггеров, тайм-аута, условий остановки), исходя из концепции эксплуатации и системных требований.

F.2.6 Структурная модель

Структурная модель системы – представление, которое показывает расположение элементов относительно друг друга и, где необходимо, показывает интерфейсы между элементами и с внешними объектами. Такая модель позволяет консолидировать или идентифицировать физические интерфейсы между элементами системы на уровне системной иерархии и между уровнями системной иерархии, а также интерфейсы с внешними объектами для соответствующей системы (в ее среде/контексте). Структурные модели могут быть иерархическими или объектно-ориентированными.

F.2.7 Сетевая модель

Сетевая модель определяет расположение узлов и ссылок, чтобы помочь понять, как ресурсы (например, информация и пользователи) перемещаются от одного узла к другому. Сетевая модель может использоваться для определения ограничений, таких как пропускная способность, время ожидания и точки перегрузки. Сетевая модель иногда моделируется вместе со стеком протоколов, чтобы понять, как уровни в сети взаимодействуют вертикально вверх и вниз по стеку.

F.3 Другие соображения по модели

Проблемы жизненного цикла заинтересованных сторон, такие как обслуживание, развитие, утилизация, потенциальные изменения среды, управление устареванием и другие нефункциональные требования, решаются путем определения архитектурных характеристик, таких как модульность, относительная независимость, масштабируемость, возможность обновления, адаптация к нескольким средам, уровень эффективности, надежности, устойчивости и отказоустойчивости. Другие необходимые модели могут включать некоторые из этих характеристик или другие критические характеристики качества. Например, случай Software Assurance, рассматриваемый как модель, может помочь в выявлении потенциальных архитектурных смягчений для минимизации операционных рисков (потеря миссии из-за использованных уязвимостей безопасности), связанных с критическими проблемами и функциями.

Определение того, какие модели использовать при определении системы, может быть основано на изучении интересов заинтересованных сторон. Модели и полученные в результате представления могут использоваться для выражения того, как архитектура и дизайн системы решают их проблемы, и для лучшего понимания их реальных потребностей, желаний и ожиданий.

Кроме того, модели могут использоваться в других процессах жизненного цикла помимо определения архитектуры и дизайна. Системное проектирование на основе моделей (MBSE) – формализованное приложение моделирования для поддержки системных требований, архитектуры, проектирования, анализа, а также операций по верификации и валидации на протяжении всего жизненного цикла.

Примечание – Модель верификации и валидации определяет представления тестовой информации, которая может поддерживать верификацию архитектуры. Модели верификации и валидации могут генерировать анализы тестов, данные, случаи и другую информацию.

# Приложение G

# (рекомендуемое)

# Применение процессов жизненного цикла программного обеспечения к системе систем

G.1 Введение

Система систем (SoS) – интересующая нас система (SOI), элементы которой сами являются системами. SoS объединяет набор систем для задачи, которую ни одна из систем не может выполнить самостоятельно. Каждая составляющая система сохраняет собственное управление, цели и ресурсы, координируя свою работу в рамках SoS и адаптируясь для достижения целей SoS. В контексте терминологии, обсуждаемой в 5.2.3 (как показано на рисунке 3), составной набор систем, включая исходную SOI, обеспечивающие системы и взаимодействующие системы, вместе составляют SoS. Когда есть проблемы, которые влияют на составной набор, система систем становится SOI, которая считается удовлетворяющей некоторой бизнес-цели или миссии, которая не может быть удовлетворена отдельными составляющими системами, или для понимания возникающего поведения комбинации.

В приложении рассматривается применение процессов жизненного цикла системы к таким SoS. Он описывает общие характеристики, общие типы SoS и последствия на протяжении жизненного цикла.

G.2 Характеристики и типы SoS

SoS характеризуются управленческой и операционной независимостью составляющих систем, которые во многих случаях были разработаны и продолжают поддерживать первоначально идентифицированных пользователей одновременно с пользователями SoS. В других контекстах каждая составляющая система сама по себе является SOI; ее существование часто предшествовало SoS, в то время как ее характеристики были изначально разработаны для удовлетворения потребностей их первоначальных пользователей. Как составные части SoS, их рассмотрение расширено, чтобы охватить более широкие потребности SoS. Это подразумевает дополнительную сложность, особенно когда системы продолжают развиваться независимо от SoS. Составляющие системы также обычно сохраняют своих первоначальных заинтересованных сторон и механизмы управления, что ограничивает альтернативы для удовлетворения потребностей SoS.

SoS были разделены на четыре типа на основе отношений управления между составляющими системами и SoS (Таблица G.1). Наиболее прочные отношения управления применяются к управляемой системе систем, где организация SoS имеет власть над составляющими системами, несмотря на то, что составляющие системы изначально не были спроектированы для поддержки SoS. Несколько меньший контроль предоставляется для признанной SoS, где распределенные полномочия между составляющими системами и системами систем влияют на применение некоторых процессов системной инженерии. В совместной SoS, в которой отсутствует система полномочий системы, применение системной инженерии зависит от взаимодействия между составляющими системами. Виртуальные системы систем в значительной степени самоорганизуются и ограничивают возможности системной инженерии SoS.

**Таблица G.1 – Система типов систем**

|  |  |
| --- | --- |
| **Тип** | **Характерная черта** |
| Виртуальная | * Децентрализованное управление * Нет явной, централизованно согласованной цели * Возникающие модели поведения, основанные на относительно невидимых механизмах непрерывности |
| Совместная | * Составляющие системы добровольно взаимодействуют для достижения согласованных целей * Коллективно решение, как взаимодействовать, обеспечивать соблюдение и поддерживать стандарты |
| Общепризнанная | * Признанные цели, назначенный менеджер и ресурсы для SoS * Составляющие системы сохраняют свое независимое владение, управление и ресурсы |
| Целевая | * Интегрированная SoS создана и управляется для выполнения конкретных задач * Централизованное управление и развитие * Составляющие системы сохраняют способность работать независимо * Нормальный режим работы подчинен центральной цели |

Ключевой характеристикой SoS является возникновение – непредвиденные эффекты на уровне SoS, приписываемые сложной динамике взаимодействия составляющих систем. Однако в SoS составные системы намеренно рассматриваются в их комбинации, чтобы получить и проанализировать результаты, которые невозможно получить с помощью одних систем. Сложность составляющих систем и тот факт, что они, возможно, были разработаны без учета их роли в SoS, могут привести к новому, неожиданному поведению. Выявление и устранение непредвиденных возникающих результатов – особая проблема в разработке SoS.

Примечание – Некоторые из крупнейших программных SoS, такие как Интернет, являются виртуальными SoS, в которых составляющие системы разработана с учетом общих рекомендаций и протоколов связи. Virtual SoS может демонстрировать полезные новые свойства поведения, такие как избыточность, динамическая реконфигурация, совместная работа и устойчивость.

G.3 Процессы системной инженерии, применяемые к системам систем

G.3.1 Общий

Вышеупомянутые характеристики SoS влияют на применение каждого из четырех типов процессов жизненного цикла системы.

G.3.2 Процессы согласования

Процессы согласования имеют решающее значение для SoS, поскольку они устанавливают режимы развития и оперативного контроля между организациями, ответственными за SoS, и часто независимыми составляющими системами. Составные системы, которые приобретаются и управляются разными организациями, иногда преследуют первоначальные цели, которые не совпадают с целями SoS. За исключением случая управляемой SoS, организация SoS не может поручить составной системной организации без их сотрудничества. В признанной или совместной SoS эти задачи сбалансированы с задачами составляющей системы как самостоятельной SOI. Для виртуальных SoS процессы согласования могут быть внутренними или рассматриваться только в целях анализа.

Даже в соглашениях между владельцами составляющих систем по-прежнему есть покупатель и поставщик. Владелец системы может быть как покупателем, так и поставщиком другой составляющей системы.

G.3.3 Организационные процессы, способствующие реализации проектов

В типичной интересующей системе процессы организационного обеспечения проекта определяют среду, в которой реализуются проекты. Организация устанавливает процессы и модели жизненного цикла, которые будут использоваться проектами; создает, перенаправляет или отменяет проекты; предоставляет необходимые ресурсы, в том числе человеческие и финансовые; и устанавливает и контролирует меры качества для систем и других результатов, которые разрабатываются проектами для внутренних и внешних клиентов (6.2).

В SoS владельцы составляющих систем обычно сохраняют ответственность за проектирование своих систем, и у каждого из них есть свои собственные процессы поддержки организационных проектов. В зависимости от типа SoS, SoS также применяет эти процессы организационного обеспечения проекта к конкретным аспектам SoS: планирование, анализ, организация и интеграция возможностей сочетания существующих и новых систем в возможности SoS.

Следовательно, в SoS эти организационные процессы поддержки проектов реализованы на двух уровнях. Организации, ответственные за составляющие системы, реализуют эти процессы для своих SOI независимо от SoS. Организация SoS (или в совместных системах систем по согласованию с SoS) реализует эти процессы для SoS с учетом тех соображений, которые относятся к общей SoS. Например, к управлению человеческими ресурсами обращается каждая системная организация, составляющая ее, для разработки своей системы. Организация SoS обращается к управлению человеческими ресурсами только для деятельности по системному проектированию, которая применяется во всех составляющих системы к SoS.

Особой проблемой в разработке SoS является отсутствие согласованности между организационными процессами поддержки проектов, составляющими систему, и процессами SoS. Составляющие системные процессы предназначены для достижения собственных результатов и иногда не согласуются с таковыми из SoS. Например, управление портфелем может быть составной системной ответственностью в тех случаях, когда составная системная организация имеет полный контроль над составной системой и другими системами и проектами в своем портфеле, в отличие от управления портфелем организации SoS.

G.3.4 Процессы технического менеджмента

В типичной интересующей системе процессы технического менеджмента связаны с управлением ресурсами и активами, выделенными руководством организации, и их применением для выполнения соглашений, в которые входит организация или организации. Они относятся к управлению проектами, в частности к планированию с точки зрения затрат, сроков и достижений, к проверке действий на соответствие планам и критериям эффективности, а также к выявлению и выбору корректирующих действий, которые восполняют недоработки и достижения. Они используются для разработки и выполнения технических планов проекта, управления информацией в технической группе, оценки технического прогресса по сравнению с планами для системных продуктов или услуг, контроля технических задач до их завершения и помощи в принятии решений процесс (6.3).

Процессы технического управления также реализуются на уровне SoS и составляющих систем. Процессы технического менеджмента применяются к конкретным аспектам проектирования SoS – планированию, анализу, организации и интеграции возможностей сочетания существующих и новых систем в возможности системы систем. Параллельно входящие в состав системные организации сохраняют ответственность за проектирование своих систем и за свои собственные процессы технического управления.

Организация SoS обращается к процессам технического управления в том виде, в каком они применяются в рамках SoS, в то время как процессы также реализуются независимо во входящих в ее состав организациях. Например, для управления конфигурацией составляющие системы управляют своими собственными конфигурациями, в то время как SoS обращается к управлению конфигурацией, поскольку это применяется к совокупности систем в SoS. Риском управляют составляющие системы на основе оценки риска применительно к их результатам, в то время как управление рисками SoS рассматривает риск для SoS.

Планирование, оценка и контроль (6.3) являются ключевыми для всех методов управления. Ключевой проблемой в разработке SoS является отсутствие контроля со стороны организации SoS над процессами для составляющих систем (особенно для признанных и совместных SoS). Каждая из составляющих систем, руководствуясь собственными организационными требованиями, может иметь график разработки или обновления, который отличается от графиков других составляющих систем. Организация SoS планирует интегрированный жизненный цикл, который распознает независимые изменения в составляющих системах, в дополнение к инициированным SoS изменениям в жизненном цикле, который рассматривает SoS как SOI. Это часто включает определение стабильных промежуточных форм, которые подчеркивают эволюцию SoS с дополнительными возможностями, добавленными среди составляющих систем.

G.3.5 Технические процессы

Технические процессы связаны с техническими действиями на протяжении всего жизненного цикла. Они сначала трансформируют потребности заинтересованных сторон в продукт, а затем, применяя этот продукт, предоставляют устойчивые услуги, когда и где это необходимо, для достижения удовлетворенности клиентов. Технические процессы применяются для создания и использования системы, будь то в форме модели или готового продукта, и они применяются на любом уровне иерархии структуры системы (6.4).

Как и в случае с другими процессами, применительно к SoS, технические процессы реализованы как для SoS, так и для составляющих систем; в некоторых случаях реализация SoS осуществляется посредством выполнения процессов, составляющих систему, а не для SoS в целом.

Анализ бизнеса или миссии для SoS охватывает всю среду бизнеса и миссии SoS. В той степени, в которой составляющая система была разработана для работы в этом пространстве, бизнес-анализ или анализ миссии для систем системы и составляющих систем будут в значительной степени совместно использоваться. Цель состоит в том, чтобы определить наилучшие средства для обеспечения желаемых возможностей.

Определение потребностей и требований заинтересованных сторон фокусируется на SoS верхнего уровня, но также рассматривает, как несопоставимые потребности заинтересованных сторон для отдельных систем могут привести к ограничениям SoS.

Определение требований к системе / программному обеспечению для SoS обычно определяется на уровне, необходимом для удовлетворения потребностей заинтересованных сторон и целей миссии, для преобразования в требования для составляющих систем, при этом SoS выступает в качестве «заинтересованного лица» для новых требований для составляющих систем.

Архитектура SoS – структура для организации и интеграции возможностей сочетания существующих и новых систем в возможности SoS, оставляя архитектуру составляющих систем их организациям. Поскольку составляющие системы в SoS обычно предшествуют SoS, определение архитектуры SoS часто начинается с фактической архитектуры SoS. Затем изучаются альтернативы архитектуры, чтобы определить заинтересованную сторону.

касается и удовлетворения требований SoS верхнего уровня, а также для распознавания влияния новых требований на составляющие системы и учета ограничений архитектуры составляющей системы.

Процесс определения дизайна предоставляет достаточно подробных данных и информации для реализации SoS. Это включает сотрудничество с входящими в состав системными организациями, которые будут проводить свои собственные разработки, чтобы определить подход к удовлетворению требований SoS, применимых к их системе. Это ответственность входящих в систему организаций. Внедрение осуществляется составляющими системами, при этом организация SoS играет роль мониторинга.

Интеграция, проверка, переход и проверка выполняются составляющими системами для изменений, которые они внедряют для поддержки требований, генерируемых SoS. Эти процессы также применимы к SoS, когда обновленные составляющие системы интегрированы в SoS, а производительность проверена и подтверждена. Независимый и асинхронный характер составляющих систем в SoS создает проблемы для эффективной реализации этих процессов, как это реализовано в традиционной SOI. В некоторых случаях оценки уровня SoS могут выполняться только в операционной среде, и в этом случае следует рассмотреть меры предосторожности, чтобы избежать неблагоприятного поведения SoS.

Наконец, процессы эксплуатации, обслуживания и утилизации, как правило, реализуются составляющими системами, учитывая их управление и операционную независимость. Взаимодействие на уровне SoS может облегчить взаимодействие и уменьшить дублирование усилий для этих процессов.

# Приложение H

# (рекомендуемое)

# Применение Agile

Документ предназначен для применения в организациях и проектах по разработке программного обеспечения, использующих гибкие подходы и методы, а также в тех, которые используют другие формальные инженерные подходы. Гибкая разработка – один из наиболее широко используемых подходов к разработке программного обеспечения (включая обслуживание программного обеспечения), поскольку он считается более доступным и позволяет быстрее предоставлять полезные продукты. В больших усилиях по разработке программного обеспечения, а также в небольших проектах, многие методы Agile могут использоваться с различными моделями жизненного цикла, и разные методы могут использоваться на разных этапах жизненного цикла. В этом приложении указаны интерпретации требований к процессу в этом документе, которые подходят для обычно используемых гибких методов.

Как обсуждалось в разделе 5.4.2, модели жизненного цикла, используемые в гибких проектах, часто являются инкрементальными и эволюционирующими. Однако организации, использующие гибкие методы, применяют процессы жизненного цикла, указанные в этом документе, включая организационное, техническое управление и технические процессы (и могут работать в рамках процессов соглашения). Как указано в 5.4.1, этот документ не предписывает какую-либо конкретную последовательность процессов в рамках модели жизненного цикла. Последовательность процессов определяется целями проекта и выбором модели жизненного цикла. В гибком проекте, поскольку он трансформирует или объединяет действия при создании или улучшении работающего программного обеспечения, может оказаться более целесообразным заявлять о полном соответствии с результатами (4.2.1), а не о полном соответствии действиям и задачам (4.2.2).

Гибкая разработка успешна отчасти из-за природы программного обеспечения, которое обеспечивает гибкость проектирования во время создания программного обеспечения. В гибкой практике проектирование программного обеспечения, реализация (конструирование) и непрерывная интеграция часто выполняются одновременно. Практика контрастирует с формальным подходом к отслеживанию сверху вниз, при котором строительство не может начаться до тех пор, пока проект не будет одобрен, так что построенное программное обеспечение прослеживается до ранее утвержденного детального проекта. Таким образом, гибкие проекты в полной мере используют подход, описанный в этом документе, в котором процессы происходят одновременно, в отличие от проектов с последовательным этапом (идеализированный водопад).

В гибких проектах исследование концепции, разработка, построение, тестирование, переход и вывод из эксплуатации предыдущего программного обеспечения могут выполняться одновременно для последовательных итераций. В Agile-проектах перепланирование часто выполняется одновременно с упомянутыми выше действиями. При таких подходах использование конца этапа в качестве контрольной точки управления или контроля не очень полезно. Другие гибкие подходы выполняют перепланирование в точках между назначенными итерациями (например, спринтами или заранее заданными временными рамками), так что каждую из этих итераций можно рассматривать как этап.

Гибкие проекты могут иметь тесно связанные циклы разработки и выпуска программного обеспечения (например, с ежедневными, еженедельными, ежемесячными запланированными выпусками) или могут отделять завершение итераций разработки от управления запланированными выпусками программного обеспечения для удобства клиента или в соответствии со стратегией организации.

Помимо применения высоко итеративной и эволюционной модели жизненного цикла, гибкие организации имеют особые практики для процессов планирования и оценки проектов, а также управления ими. Вместо того, чтобы устанавливать основные контрольные точки при переходе между этапами или процессами, гибкие проекты часто проводят менее формальные контрольные точки или ретроспективные обзоры в конце ограниченного по времени цикла, чтобы согласовать улучшения для следующего цикла. Каждая итерация включает действия по проектированию, разработке и тестированию (разработка через тестирование). После спринта продолжительностью от одной до четырех недель или дольше новые рабочие элементы программного обеспечения принимаются как «готовые» - полностью разработанные, проверенные (протестированные) и утвержденные. Выявляются извлеченные уроки и улучшения процесса, и начинается работа над еще одним спринтом. Постоянное обучение, управление рисками.

Гибкие методы делают упор на процессе определения потребностей и требований заинтересованных сторон, способствуя изменениям за счет высокой степени постоянного участия заинтересованных сторон. В гибких проектах ключевые заинтересованные стороны, такие как покупатель или представители пользователей, не только утверждают информацию, измерения и отчеты об оценке. Непрерывное участие заинтересованных сторон согласуется с этим документом, который определяет участие заинтересованных сторон в каждом техническом процессе, а также в процессе адаптации (Приложение A). Они активно участвуют в управлении требованиями (6.4.3.3.d) на каждой итерации, внося новые требования и изменяя приоритеты и участвуя, когда приоритетные требования выбираются из неразвитых историй или функций и дорабатываются для разработки. Итерационный подход поощряет гибкость добавления, требования, которые признаны входящими в общий объем проекта. Кроме того, участие заинтересованных сторон в утверждении тестируемого программного обеспечения на каждой итерации означает, что проверка проводится на протяжении всего проекта.

С постепенным определением эволюционирующих требований концепция содержания проекта отличается в гибком проекте от проектов, в которых содержание определяется заранее определенной базовой линией указанных требований. Когда Agile-проекту требуется определенный продукт, его объем изначально привязан к высокоуровневым или фундаментальным требованиям. Ожидается, что более подробные уровни определения продукта появятся по мере получения дополнительных знаний в процессе строительства. Гибкая работа без заранее определенных продуктов (например, поддержание уровня усилий) может контролировать объем с помощью ограниченных по времени графиков или групп с ограниченными ресурсами. Этот подход особенно подходит для работы по сопровождению программного обеспечения, когда объем или содержание корректирующих или адаптивных работ не полностью определены изначально.

Спецификация исходных условий также отличается по степени и срокам для проектов гибкой разработки по сравнению с более традиционными усилиями по разработке. Базовый уровень требований может изначально включать пользовательские истории высокого уровня («эпики») и ключевые измерения производительности, включая стандарты удобства использования. Гибкие проекты в полной мере используют задачу «Определение базовых показателей на протяжении жизненного цикла» (6.3.5.3.b) 4). Во время разработки новые базовые планы согласовываются и строятся, по крайней мере, ежедневно под контролем конфигурации. Программный элемент, как правило, отслеживается до функциональных требований высокого уровня, а также до варианта использования, который он реализует, и тестового примера, используемого для проверки его функциональности и производительности вместо того, чтобы быть привязанным к ранее утвержденному базовому проектному документу.

Подготовка спецификаций, артефактов дизайна и элементов информации или документации во время гибких проектов часто ограничена, в то время как разработчики программного обеспечения тратят свое время и навыки на преобразование сценария или описания функции («пользовательской истории») в работающую, поддающуюся тестированию. Программный элемент вместо того, чтобы готовить тщательно продуманные пакеты проверок для брифингов при нечастых проверках основных этапов, команда часто встречается с заинтересованными сторонами, чтобы представить достоверные доказательства выполнения набора функций и согласовать содержание следующей итерации. Документированные элементы информации касаются того, что потребуется для перехода, эксплуатации и обслуживания, например документации для операторов и конечных пользователей, а также базовых версий протестированных и выпущенных версий программного обеспечения с планами тестирования и тестовыми примерами. В проектах повторно используются организационные процедуры для управления конфигурацией и выпусками, проверки, а также управления инцидентами и проблемами. Там, где это возможно, двунаправленная прослеживаемость обеспечивается и обеспечивается интегрированными автоматизированными системами и процедурами для управления требованиями, архитектуры и дизайна, управления конфигурацией, измерениями и управлением информацией.

Инкрементальный и итеративный характер Agile-разработки может способствовать эффективным техническим и управленческим процессам и методам для снижения затрат, связанных с изменениями. Для сравнения, проекты, управляемые на каскадном этапе континуума, стремятся снизить общую стоимость доработки за счет минимизации количества изменений, ограничения количества контрольных точек и определения базовых подробных спецификаций, которые проверяются и отслеживаются на протяжении всего проекта.

Гибкие проекты для сложных систем пытаются управлять затратами, отдавая приоритет наиболее важным возможностям для ранней реализации. Если организация управляет разработкой и обслуживанием всего своего портфеля программных систем как единой системы, управляемой уровнем затрат, а не общими затратами, то организация может, в принципе, управлять этим портфелем систем как непрерывным гибким развитием, аналогично управление высокотехнологичным итеративным обслуживанием «Kanban» (Канбан).

# Приложение G

# (рекомендуемое)

# Отображение процессов в соответсвии с ISO/IEC/IEEE 12207: 2008

В документе используется модель процесса, идентичная [ISO/IEC/IEEE 15288: 2015]. Чтобы обеспечить прослеживаемость и упростить переход для пользователей предыдущей (2008 г.) редакции [ISO/IEC/IEEE 12207], в таблице I.1 представлена перекрестная ссылка на процессы, которая показывает согласование основных процессов между двумя версиями. Таблица I.1 не означает, что определения процессов в редакциях 2008 и 2017 гг. данного документа идентичны. В некоторых случаях процессы в более ранней версии теперь решаются посредством действий и задач или описываются в примечаниях. Возможны другие, более подробные сопоставления для согласования цели процесса, результатов, действий или задач двух версий. Одно такое отображение показано в таблице I.2.

**Таблица I.1 - Сравнение процессов в [ISO/IEC/IEEE 12207: 2017] и предыдущей редакции**

|  |  |  |  |
| --- | --- | --- | --- |
| **ISO/IEC/IEEE 12207: 2017 и ISO/IEC/IEEE**  **15288: 2015 процесс** | | **ISO/IEC 12207: 2008 (IEEE Std 12207: 2008) процесс** | |
| **Подпункт** | **Заголовок** | **Подпункт** | **Заголовок** |
| 6 Программное обеспечение процессов жизненного цикла | | 1. Процессы жизненного цикла системы 2. Процессы, специфичные для программного обеспечения | |
| 6.1 Процессы согласования | | 6.1 Процессы согласования | |
| 6.1.1 Процесс приобретения | | 6.1.1 Процесс приобретения | |
| 6.1.2 Процесс поставки | | 6.1.2 Процесс поставки | |
| 6.2 Организационные процессы, способствующие реализации проектов | | 6.2 Организационные процессы, способствующие реализации проектов | |
| 6.2.1 Процесс управления моделью жизненного цикла | | 6.2.1 Процесс управления моделью жизненного цикла | |
| 6.2.2 Процесс управления инфраструктурой | | 6.2.2 Процесс управления инфраструктурой | |
| 6.2.3 Процесс управления проектным портфелем | | 6.2.3 Процесс управления портфелем проектов | |
| 6.2.4 Процесс управления человеческими ресурсами | | 6.2.4 Процесс управления человеческими ресурсами | |
| 6.2.5 Качественный процесс управления | | 6.2.5 Качественный Процесс управления | |
| 6.2.6 Процесс управления знаниями | | 6.2.4.2.e Управление человеческими ресурсами  6.2.4.3.4 Управление знаниями (деятельность)   * 1. Процессы повторного использования программного обеспечения      1. Процесс разработки домена      2. Процесс управления активами повторного использования      3. Процесс управления программами повторного использования | |
| 6.3 Процессы технического менеджмента | | 6.3 Процессы проекта  7.2 Процессы поддержки программного обеспечения | |
| 6.3.1 Процесс планирования проекта | | 6.3.1 Процесс оценки и контроля проекта | |
| 6.3.2 Процесс оценки и контроля проекта | | 6.3.2 Процесс оценки и контроля проекта  7.2.6 Процесс рецензирования программного обеспечения | |
| 6.3.3 Процесс принятия решений | | 6.3.3 Процесс управления решениями | |
| 6.3.4 Процесс управления рисками | | 6.3.4 Процесс управления рисками | |
| 6.3.5 Процесс управления конфигурацией | | 6.3.5 Конфигурация процесса управления  7.2.2 Процесс управления конфигурацией программного обеспечения | |

**Окончание таблицы I.1**

|  |  |  |  |
| --- | --- | --- | --- |
| **ISO/IEC/IEEE 12207: 2017 и ISO/IEC/IEEE**  **15288: 2015 процесс** | | **ISO/IEC 12207: 2008 (IEEE Std 12207: 2008) процесс** | |
| **Подпункт** | **Заголовок** | **Подпункт** | **Заголовок** |
| 6.3.6 Процесс управления информацией | | 6.3.6 Процесс управления информацией  7.2.1 Процесс управления программной документацией | |
| 6.3.7 Процесс измерения | | 6.3.7 Процесс измерения | |
| 6.3.8 Качественный процесс гарантии | | 7.2.3 Процесс обеспечения качества программного обеспечения   * + 1. Процесс аудита программного обеспечения     2. Процесс разрешения проблем программного обеспечения | |
| 6.4 Технические процессы | | 6.4 Технические процессы | |
| 6.4.1 Процесс анализа бизнеса или цели работы | | Неопределено, [но связано с некоторыми результатами 6.4.1 процесса определения потребностей и требований заинтересованных сторон] | |
| 6.4.2 Процесс определения потребностей и требований заинтересованных сторон | | 6.4.1 Процесс определения требований заинтересованных сторон | |
| 6.4.3 Процесс определения требований к системе/программному обеспечению | | 6.4.2 Процесс анализа системных требований  7.1.2 Процесс анализа требований к программному обеспечению | |
| 6.4.4 Процесс определения архитектуры | | 6.4.3 Процесс архитектурного проектирования системы  7.1.3 Процесс архитектурного проектирования программного обеспечения | |
| 6.4.5 Процесс определения дизайна | | 6.4.3 Процесс архитектурного проектирования системы  7.1.4 Процесс детального проектирования программного обеспечения | |
| 6.4.6 Процесс системного анализа | | Неопределено, [но деятельность по анализу присутствует во многих процессах, особенно в процессе управления решениями] | |
| 6.4.7 Реализация процесс | | 6.4.4 Процесс внедрения   * 1. Процессы внедрения программного обеспечения      1. Процесс внедрения программного обеспечения   7.1.5 Процесс конструирования программного обеспечения | |
| 6.4.8 Процесс интеграции | | 6.4.5 Процесс системной интеграции  7.1.6 Процесс интеграции программного обеспечения | |
| 6.4.9 Процесс верификации | | 6.4.6 Процесс квалификационного тестирования системы  7.1.7 Процесс квалификационного тестирования программного обеспечения  7.2.4 Процесс верификации программного обеспечения | |
| 6.4.10 Процесс перехода | | 6.2.4 Процесс управления человеческими ресурсами  6.4.7 Процесс установки программного обеспечения | |
| 6.4.11 Процесс валидации | | 6.4.8 Процесс поддержки приемки программного обеспечения  7.2.5 Процесс валидации программного обеспечения | |
| 6.4.12 Процесс эксплуатации | | 6.4.9 Процесс эксплуатации программного обеспечения | |
| 6.4.13 Процесс технического обслуживания | | 6.4.10 Процесс сопровождения программного обеспечения | |
| 6.4.14 Процесс утилизации | | 6.4.11 Процесс утилизации программного обеспечения | |
| Приложение A Процесс адаптации | | Приложение A Процесс адаптации | |

В таблице I.2 представлено сопоставление результатов процессов в этом документе с результатами выбранных программных процессов реализации, повторного использования и поддержки в ISO/IEC/IEEE 12207: 2008. Пользователи 12207: 2008 могут определить результаты этого документа, связанные с предыдущими версиями процессов, действий или задач более низкого уровня, связанных с программным обеспечением, для реализации, поддержки или повторного использования элементов программного обеспечения в системе программного обеспечения. Таблица I.2 не подразумевает, что результаты процессов идентичны в редакциях этого документа 2008 и 2017 гг. Не все результаты процессов этого документа явно указаны в предыдущем издании.

**Таблица I.2 - Сравнение результатов процесса в [ISO/IEC/IEEE 12207: 2017] и результатов, связанных с программным обеспечением, в предыдущей редакции**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Результат процесса ISO/IEC/IEEE 12207: 2017 и ISO/IEC/IEEE 15288: 2015** | | **Результат процесса ISO/IEC 12207: 2008 (IEEE Std 12207-2008)** | | |
| **Подпункт** | **Результат** | **Подпункт** | | **Результат** |
| 6.2.6 Процесс управления знаниями | | 7.3.1 Процесс разработки домена | | |
| 6.2.6.2.a | Определена таксономия для применения активов знаний. | 7.3.1.2.a,  7.3.1.2.b,  7.3.1.2.c,  7.3.1.2.d | | Выбираются формы представления для моделей домена и архитектуры домена.  Устанавливаются границы домена и его отношения с другими доменами.  Разрабатывается модель домена, которая отражает основные общие и различные характеристики, возможности, концепции и функции в домене.  Разрабатывается архитектура домена, описывающая семейство систем в домене, включая их общие черты и различия. |
| 6.2.6.2.b | Развиваются или приобретаются организационные знания, навыки и активы знаний. | 7.3.1.2.e,  7.3.1.2.f | | Активы, принадлежащие домену, определены.  Активы, принадлежащие домену, приобретаются или разрабатываются и поддерживаются в течение всего жизненного цикла. |
| 6.2.6.2.c | Имеются организационные знания, навыки и активы знаний. | 7.3.1.2.g | | Модели домена и архитектуры поддерживаются на протяжении всего жизненного цикла. |
|  | | 7.3.2 Процесс управления активами повторного использования | | |
| 6.2.6.2.a | Определена таксономия для применения активов знаний. | 7.3.2.2.a,  7.3.2.2.b | | Документируется стратегия управления активами.  Создана схема классификации активов. |
| 6.2.6.2.b | Развиваются или приобретаются организационные знания, навыки и активы знаний. | 7.3.2.2.c | | Определены критерии приемки, сертификации и списания активов. |
| 6.2.6.2.c | Доступны организационные знания, навыки и активы знаний. | 7.3.2.2.d,  7.3.2.2.f,  7.3.2.2.g | | Работает механизм хранения и поиска активов.  Изменения в активах контролируются.  Пользователи активов уведомляются об обнаруженных проблемах, внесенных изменениях, создании новых версий и удалении активов из механизма хранения и поиска. |
| 6.2.6.2.d | Собираются и анализируются данные об использовании управления знаниями. | 7.3.2.2.e | | Использование активов отражается в учете. |
|  | | 7.3.3 Процесс управления программой повторного использования | | |
| 6.2.6.2.a | Определена таксономия для применения активов знаний. | 7.3.3.2.a,  7.3.3.2.b | | Определяется стратегия повторного использования организации, включая ее цель, сферу применения, цели и задачи. |
| **Результат процесса ISO/IEC/IEEE 12207: 2017 и ISO/IEC/IEEE 15288: 2015** | | | **Результат процесса ISO/IEC 12207: 2008 (IEEE Std 12207-2008)** | |
| **Подпункт** | **Результат** | | **Подпункт** | **Результат** |
|  | | | Определяются домены для потенциальных возможностей повторного использования. | |
| 6.2.6.2.b | Организационные знания, навыки и активы знаний разработаны или приобретены. | | 7.3.3.2.c,  7.3.3.2.d,  7.3.3.2.e | Оценивается способность организации к систематическому повторному использованию.  Оценивается потенциал повторного использования в каждой области.  Оцениваются предложения по повторному использованию, чтобы убедиться, что продукт повторного использования подходит для предлагаемого применения. |
| 6.2.6.2.c | Организационные знания, навыки и активы знаний доступны. | | 7.3.3.2.f | Стратегия повторного использования внедряется в организации. |
| 6.2.6.2.d | Данные об использовании управления знаниями собираются и анализируются. | | 7.3.3.2.g,  7.3.3.2h | Созданы механизмы обратной связи, коммуникации и уведомления, которые действуют между затронутыми сторонами.  Программа повторного использования контролируется и оценивается. |
| 6.3.1 Процесс планирования проекта | | | 7.2.6 Процесс рецензирования программного обеспечения | |
| 6.3.1.2.a | Определены цели и планы. | | 7.2.6.2.a | Управленческие и технические обзоры проводятся в соответствии с потребностями проекта. |
| 6.3.1.2.b | Определены роли, обязанности, ответственность и полномочия. | | 7.2.6.2.a | Управленческие и технические обзоры проводятся в соответствии с потребностями проекта. |
| 6.3.1.2.c | Ресурсы и услуги, необходимые для достижения целей, официально запрошены и предоставлены. | | 7.2.6.2.a | Управленческие и технические обзоры проводятся в соответствии с потребностями проекта. |
| 6.3.1.2.d | Планы по реализации проекта активированы. | | 7.2.6.2.a | Управленческие и технические обзоры проводятся в соответствии с потребностями проекта. |
| 6.3.2 Процесс планирования проекта | | | 7.2.6 Процесс рецензирования программного обеспечения | |
| 6.3.2.2.a | Доступны показатели эффективности или результаты оценки. | | 7.2.6.2.c,  7.2.6.2.e | Результаты анализа доводятся до сведения всех заинтересованных сторон.  Риски и проблемы выявляются и регистрируются. |
| 6.3.2.2.b | Оценивается адекватность ролей, ответственности, ответственности и полномочий. | | 7.2.6.2a  7.2.6.2.b | Управленческие и технические обзоры проводятся в соответствии с потребностями проекта.  Статус и результаты деятельности процесса оцениваются посредством проверок. |
| 6.3.2.2.c | Оценивается достаточность ресурсов | | 7.2.6.2.a  7.2.6.2.b | Управленческие и технические обзоры проводятся в зависимости от потребностей проекта.  Статус и продукты деятельности в рамках процесса оцениваются посредством обзорных мероприятий. |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Результат процесса ISO/IEC/IEEE 12207: 2017 и ISO/IEC/IEEE 15288: 2015** | | | | | **Результат процесса ISO/IEC 12207: 2008 (IEEE Std 12207-2008)** | | | |
| **Подпункт** | | | **Результат** | | **Подпункт** | | | **Результат** |
| 6.3.2.2.d | | | Выполняются обзоры технического прогресса. | | 7.2.6.2.a | | | Управленческие и технические обзоры проводятся в соответствии с потребностями проекта. |
| 6.3.2.2.e | | | Исследуются и анализируются отклонения в исполнении проекта от планов. | | 7.2.6.2.b | | | Состояние и продукты деятельности в рамках процесса оцениваются с помощью мероприятий по анализу. |
| 6.3.2.2.f | | | Заинтересованные стороны информируются о состоянии проекта. | | 7.2.6.2.c | | | Результаты анализа доводятся до сведения всех заинтересованных сторон. |
| 6.3.2.2.g | | | Определяются и направляются корректирующие действия, если результаты проекта не соответствуют целевым показателям. | | 7.2.6.2.d | | | Пункты действий, возникающие в результате анализа, отслеживаются до их завершения. |
| 6.3.4 Процесс управления рисками | | | 7.2.6 Процесс обзора программного обеспечения | |  | | |  |
| 6.3.4.2.a | | | Выявлены риски. | | 7.2.6.2.e | | | Риски и проблемы выявляются и регистрируются. |
| 6.3.5 Процесс управления конфигурацией | | | | | 7.2.2 Управление конфигурацией программного обеспечения | | | |
| 6.3.5.2.a | | | Элементы, требующие управления конфигурацией, идентифицируются и управляются. | | 7.2.2.2.a,  7.2.2.2.b | | | Разработана стратегия управления конфигурацией программного обеспечения.  Элементы, созданные в процессе или проекте, идентифицируются, определяются и базируются на них. |
| 6.3.5.2.b | | | Базовые параметры конфигурации установлены. | | 7.2.2.2.b | | | Элементы, созданные в процессе или проекте, идентифицируются, определяются и базируются на них. |
| 6.3.5.2.c | | | Изменения в элементах управления конфигурацией находятся под контролем. | | 7.2.2.2.c,  7.2.2.2.d | | | Модификации и выпуски элементов находятся под контролем.  Изменения и выпуски доступны затронутым сторонам. |
| 6.3.5.2.d | | | Информация о состоянии конфигурации доступна. | | 7.2.2.2.e | | | Состояние элементов и модификаций записывается и сообщается. |
| 6.3.5.2.e | | | Необходимые аудиты конфигурации завершены. | | 7.2.2.2.f | | | Обеспечивается комплектность и непротиворечивость позиций. |
| 6.3.5.2.f | | | Релизы и поставки системы контролируются и утверждаются.. | | 7.2.2.2.c,  7.2.2.2.d,  7.2.2.2.g | | | Модификации и релизы элементов контролируются.  Модификации и релизы предоставляются заинтересованным сторонам.  Хранение, обработка и доставка предметов контролируются. |
| 6.3.6 Процесс управления информацией | | | | | 7.2.1 Процесс управления программной документацией | | | |
| 6.3.6.2.a | | | Определяется информация, которой необходимо управлять. | | 7.2.1.2.a,  7.2.1.2.c | | | Разработана стратегия, определяющая документацию, которая должна быть подготовлена в течение жизненного цикла программного продукта или услуги.  Определяется документация, которая должна быть подготовлена в рамках процесса или проекта. |
| 6.3.6.2.b | | | Определяются представления информации. | | 7.2.1.2.b,  7.2.1.2.d | | | Определены стандарты, которые должны применяться для разработки документации программного обеспечения.  Содержание и назначение всей документации уточняется, рассматривается и утверждается. |
| 6.3.6.2.c | | | Информация получена, разработана, преобразована, сохранена, проверена, представлена и утилизирована. | | 7.2.1.2.d,  7.2.1.2.e,  7.2.1.2.f | | | Содержание и назначение всей документации уточняется, рассматривается и утверждается.  Документация разрабатывается и предоставляется в соответствии с установленными стандартами.  Документация ведется в соответствии с определенными критериями. |
| 6.3.6.2.d | | | Определяется статус информации. | | 7.2.1.2.f | | | Документация ведется в соответствии с определенными критериями. |
| 6.3.6.2.e | | | Информация доступна назначенным заинтересованным сторонам. | | 7.2.1.2.e | | | Документация разрабатывается и предоставляется в соответствии с установленными стандартами. |
| 6.3.8 Процесс обеспечения качества | | | | | 7.2.3 Процесс обеспечения качества программного обеспечения | | | |
| 6.3.8.2.a | | Определены и внедрены процедуры обеспечения качества проекта. | | | 7.2.3.2.a,  7.2.3.2.d | | Разработана стратегия обеспечения качества программного обеспечения.  Проверяется соответствие программных продуктов, процессов и действий применимым стандартам, процедурам и требованиям. | |
| 6.3.8.2.b | | Определены критерии и методы оценки обеспечения качества. | | | 7.2.3.2.a | | Разработана стратегия обеспечения качества программного обеспечения. | |
| 6.3.8.2.c | | Оценка продуктов, услуг и процессов проекта выполняется в соответствии с политиками, процедурами и требованиями менеджмента качества. | | | 7.2.3.2.d | | Проверяется соответствие программных продуктов, процессов и действий применимым стандартам, процедурам и требованиям. | |
| 6.3.8.2.d | | Результаты оценки предоставляются соответствующим заинтересованным сторонам. | | | 7.2.3.2.b | | Создаются и поддерживаются свидетельства обеспечения качества программного обеспечения. | |
| 6.3.8.2.f | | Решаются приоритетные проблемы. | | | 7.2.3.2.c | | Проблемы и / или несоответствие требованиям выявляются и регистрируются. | |
|  | | | | | 7.2.6 Процесс рецензирования программного обеспечения | | | |
| 6.3.8.2.a | | Определены и внедрены процедуры обеспечения качества проекта. | | | 7.2.6.2a | | Управленческие и технические обзоры проводятся в соответствии с потребностями проекта. | |
| 6.3.8.2.b | | Определены критерии и методы оценки обеспечения качества. | | | 7.2.6.2a | | Управленческие и технические обзоры проводятся в соответствии с потребностями проекта. | |
| 6.3.8.2.c | | Оценка продуктов, услуг и процессов проекта выполняется в соответствии с политиками, процедурами и требованиями менеджмента качества. | | | 7.2.6.2.a,  7.2.6.2.b | | Управленческие и технические обзоры проводятся в соответствии с потребностями проекта.  Статус и результаты деятельности процесса оцениваются посредством проверок. | |
| 6.3.8.2.d | | Результаты оценки предоставляются соответствующим заинтересованным сторонам. | | | 7.2.6.2.c | | Результаты проверки доводятся до сведения всех заинтересованных сторон. | |
| 6.3.8. 2.f | | Решаются приоритетные проблемы. | | | 7.2.6.2.d,  7.2.6.2.e | | Действия, возникающие в результате обзоров, отслеживаются до закрытия.  Риски и проблемы выявляются и регистрируются. | |
|  | | | | | 7.2.7 Процесс аудита программного обеспечения | | | |
| 6.3.8.2.a | | Определены и внедрены процедуры обеспечения качества проекта. | | | 7.2.7.2.a | | Разработана и внедрена стратегия аудита. | |
| 6.3.8.2.b | | Определены критерии и методы оценки обеспечения качества. | | | 7.2.7.2.a | | Разработана и внедрена стратегия аудита. | |
| 6.3.8.2.c | | Оценка продуктов, услуг и процессов проекта выполняется в соответствии с политиками, процедурами и требованиями менеджмента качества. | | | 7.2.7.2.b,  7.2.7.2.c | | Соответствие выбранных программных рабочих продуктов и / или услуг или процессов требованиям, планам и соглашению определяется в соответствии со стратегией аудита.  Аудиты проводятся соответствующей независимой стороной. | |
| 6.3.8.2.d | | Результаты оценки предоставляются соответствующим заинтересованным сторонам. | | | 7.2.7.2.d | | Проблемы, обнаруженные во время аудита, выявляются и сообщаются лицам, ответственным за корректирующие действия и решение. | |
|  | | | | | 7.2.8 Процесс разрешения проблем с программным обеспечением | | | |
| 6.3.8.2.a | | Определены и внедрены процедуры обеспечения качества проекта. | | | 7.2.8.2.a | | Разработана стратегия управления проблемами. | |
| 6.3.8.2.b | | Определены критерии и методы оценки обеспечения качества. | | 7.2.8.2.a | | | Разработана стратегия обеспечения качества программного обеспечения. | |
| 6.3.8.2.d | | Результаты оценки предоставляются соответствующим заинтересованным сторонам. | | 7.2.8.2.f | | | Статус всех обнаруженных проблем известен. | |
| 6.3.8.2.e | | Инциденты разрешены. | |  | | |  | |
| 6.3.8.2.f | | Решаются приоритетные проблемы. | | 7.2.8.2.b,  7.2.8.2.c,  7.2.8.2.d,  7.2.8.2.e | | | Проблемы регистрируются, выявляются и классифицированы.  Проблемы анализируются и оцениваются для определения приемлемого решения/решений.  Реализовано решение проблемы.  Проблемы отслеживаются до их устранения. | |
| 6.4.3 Процесс определения требований к системе/ программному обеспечению | | | | 7.1.2 Процесс анализа требований к программному обеспечению | | | | |
| 6.4.3.2.a | | Определяется описание системы или элемента, включая интерфейсы, функции и границы, для системного решения. | | 7.1.2.2.a | | | Определены требования, предъявляемые к программным элементам системы и их интерфейсам. | |
| 6.4.3.2.b | | Определяются системные/ программные требования (функциональные, производственные, процессные, нефункциональные и интерфейсные) и проектные ограничения. | | 7.1.2.2.a,  7.1.2.2.e,  7.1.2.2.f,  7.1.2.2.h | | | Определены требования, предъявляемые к программным элементам системы и их интерфейсам.  Определяется приоритетность реализации требований к программному обеспечению.  Требования к программному обеспечению утверждаются и обновляются по мере необходимости.  Требования к программному обеспечению определены и доведены до сведения всех затронутых сторон. | |
| 6.4.3.2.c | | Определены критические показатели эффективности. | | 7.1.2.2.c | | | Понятно влияние требований к программному обеспечению на операционную среду. | |
|  | |  | |  | | |  | |
| 6.4.3.2.d | | Анализируются системные / программные требования. | | 7.1.2.2.b,  7.1.2.2.c,  7.1.2.2.g | | | Требования к программному обеспечению анализируются на корректность и тестируемость.  Понятно влияние требований к программному обеспечению на операционную среду.  Изменения требований к программному обеспечению оцениваются по стоимости, графику и техническому влиянию. | |
| 6.4.3.2.e | | Доступны любые вспомогательные системы или услуги, необходимые для определения требований к системе/программному обеспечению. | |  | | | | |
| 6.4.3.2.f | | Обеспечивается отслеживаемость требований к системе/ программному обеспечению в соответствии с требованиями заинтересованных сторон. | | 7.1.2.2.d | | | Согласованность и прослеживаемость устанавливаются между требованиями к программному обеспечению и требованиями системы. | |
| 6.4.4 Процесс определения архитектуры | | | | 7.1.3 Процесс архитектурного проектирования программного обеспечения | | | | |
| 6.4.4.2.a | | Выявленные проблемы заинтересованных сторон решаются архитектурой. | | 7.1.3.2.a,  7.1.3.2.c | | | Разрабатывается и обосновывается архитектурный проект программного обеспечения, который описывает элементы программного обеспечения, реализующие требования к программному обеспечению.  Согласованность и прослеживаемость устанавливаются между требованиями к программному обеспечению и проектом программного обеспечения. | |
| 6.4.4.2.b | Развиваются точки зрения на архитектуру. | | | 7.1.3.2.a | | Разработан и составлен базовый проект архитектуры программного обеспечения, который описывает элементы программного обеспечения, которые будут реализовывать требования к программному обеспечению. | | |
| 6.4.4.2.c | Определены контекст, границы и внешние интерфейсы системы. | | | 7.1.3.2.b | | Определены внутренние и внешние интерфейсы каждого программного элемента. | | |
| 6.4.4.2.d | Разработаны архитектурные представления и модели системы. | | | 7.1.3.2.a | | Разработан и составлен базовый проект архитектуры программного обеспечения, который описывает элементы программного обеспечения, которые будут реализовывать требования к программному обеспечению. | | |
| 6.4.4.2.e | Концепции, свойства, характеристики, поведение, функции или ограничения, которые важны для архитектурных решений системы, назначаются архитектурным объектам. | | | 7.1.3.2.a | | Разработан и составлен базовый проект архитектуры программного обеспечения, который описывает элементы программного обеспечения, которые будут реализовывать требования к программному обеспечению. | | |
| 6.4.4.2.f | Идентифицируются элементы системы и их интерфейсы. | | | 7.1.3.2.a,  7.1.3.2.b | | Разработан и составлен базовый проект архитектуры программного обеспечения, который описывает элементы программного обеспечения, которые будут реализовывать требования к программному обеспечению.  Определены внутренние и внешние интерфейсы каждого программного элемента. | | |
| 6.4.4.2.g | Оцениваются варианты архитектуры. | | | 7.1.3.2.a | | Разработан и составлен базовый проект архитектуры программного обеспечения, который описывает элементы программного обеспечения, которые будут реализовывать требования к программному обеспечению. | | |
| 6.4.4.2.h | Создается архитектурная основа для процессов на протяжении всего жизненного цикла. | | | 7.1.3.2.a | | Разработан и составлен базовый проект архитектуры программного обеспечения, который описывает элементы программного обеспечения, которые будут реализовывать требования к программному обеспечению. | | |
| 6.4.4.2.i | Достигнуто соответствие архитектуры требованиям и конструктивным характеристикам. | | | 7.1.3.2.a,  7.1.3.2.c | | Разработан и составлен базовый проект архитектуры программного обеспечения, который описывает элементы программного обеспечения, которые будут реализовывать требования к программному обеспечению.  Согласованность и прослеживаемость устанавливаются между требованиями к программному обеспечению и проектированием программного обеспечения. | | |
| 6.4.4.2.j | Доступны любые вспомогательные системы или услуги, необходимые для определения архитектуры. | | |  | | | | |
| 6.4.4.2.k | Обеспечивается прослеживаемость элементов архитектуры к заинтересованным сторонам и требованиям системы/ программного обеспечения. | | | 7.1.3.2.c | | Согласованность и прослеживаемость устанавливаются между требованиями к программному обеспечению и проектированием программного обеспечения. | | |
|  | | | | 7.3.1 Процесс проектирования домена | | | | |
| 6.4.4.2.c | Определены контекст, границы и внешние интерфейсы системы. | | | 7.3.1.2.b | | Устанавливаются границы домена и его отношения с другими доменами. | | |
| 6.4.4.2.d | Разработаны архитектурные представления и модели системы. | | | 7.3.1.2.a,  7.3.1.2.c | | Выбираются формы представления моделей предметной области и архитектуры предметной области.  Разрабатывается модель предметной области, которая фиксирует основные общие и различные функции, возможности, концепции и функции предметной области. | | |
| 6.4.4.2.h | | Создается архитектурная основа для процессов на протяжении всего жизненного цикла. | | 7.3.1.2.d | | | Разрабатывается доменная архитектура, описывающая семейство систем в домене, включая их общие черты и вариативность. | |
| 6.4.5 Процесс определения дизайна | | | | 7.1.4 Процесс детального проектирования программного обеспечения | | | | |
| 6.4.5.2.a | | Определены конструктивные характеристики каждого элемента системы. | | 7.1.4.2.a | | | Разрабатывается подробный проект каждого программного компонента с описанием создаваемых программных модулей. | |
| 6.4.5.2.b | | Системные/программные требования распределяются по системным элементам. | | 7.1.4.2.a,  7.1.4.2.c | | | Разрабатывается подробный проект каждого программного компонента с описанием создаваемых программных модулей.  Согласованность и прослеживаемость устанавливаются между детальным проектом и требованиями и архитектурным проектом. | |
| 6.4.5.2.c | | Выбираются или определяются факторы, способствующие проектированию, необходимые для определения проекта. | | 7.1.4.2.a | | | Разрабатывается подробный проект каждого программного компонента с описанием создаваемых программных модулей. | |
| 6.4.5.2.d | | Определяются или уточняются интерфейсы между элементами системы, составляющими систему. | | 7.1.4.2.b | | | Определены внешние интерфейсы каждого программного модуля. | |
| 6.4.5.2.e | | Оцениваются варианты конструкции элементов системы. | | 7.1.4.2.a | | | Разрабатывается подробный проект каждого программного компонента с описанием создаваемых программных модулей. | |
| 6.4.5.2.f | | Разрабатываются ошибки дизайна. | | 7.1.4.2.a | | | Разрабатывается подробный проект каждого программного компонента с описанием создаваемых программных модулей. | |
| 6.4.5.2.g | | Доступны любые вспомогательные системы или услуги, необходимые для определения дизайна. | |  | | | | |
| 6.4.5.2.h | | Установлена прослеживаемость проектных характеристик к архитектурным объектам архитектуры системы. | | 7.1.4.2.c | | | Согласованность и прослеживаемость устанавливаются между детальным проектом и требованиями и архитектурным проектом. | |
| 6.4.7 Процесс внедрения | | | | 7.1.1 Процесс внедрения программного обеспечения | | | | |
| 6.4.7.2.a | | Выявлены ограничения реализации, влияющие на требования, архитектуру или дизайн. | | 7.1.1.2.a,  7.1.1.2.b | | | Определена стратегия реализации.  Выявлены технологические ограничения реализации проекта. | |
| 6.4.7.2.b | | Реализован системный элемент. | | 7.1.1.2.c | | | Реализован программный продукт. | |
| 6.4.7.2.c | | Системный элемент упаковывается или хранится. | | 7.1.1.2.d | | | Программный объект упаковывается и хранится в соответствии с договором на его поставку. | |
| 6.4.7.2.d | | Доступны любые вспомогательные системы или услуги, необходимые для реализации. | |  | | | | |
|  | | | | 7.1.5 Процесс создания программного обеспечения | | | | |
| 6.4.7.2.a | | Выявлены ограничения реализации, влияющие на требования, архитектуру или дизайн. | | 7.1.5.2.a,  7.1.5.2.d | | | Критерии проверки определены для всех программных модулей на соответствие их требованиям.  Проверка программных модулей на соответствие требованиям и дизайн завершены. | |
| 6.4.7.2.b | Реализован системный элемент. | | | 7.1.5.2.b,  7.1.5.2.d | | Производятся единицы программного обеспечения, определенные проектом.  Проверка программных модулей на соответствие требованиям и дизайн завершены. | | |
| 6.4.7.2.e | Прослеживаемость установлена. | | | 7.1.5.2.c | | Согласованность и прослеживаемость устанавливаются между программными единицами, требованиями и дизайном. | | |
| 6.4.8 Процесс интеграции | | | | 7.1.6 Процесс интеграции программного обеспечения | | | | |
| 6.4.8.2.a | Выявлены ограничения интеграции, которые влияют на системные требования, архитектуру или дизайн, включая интерфейсы. | | | 7.1.6.2.a | | Для модулей программного обеспечения разрабатывается стратегия интеграции, соответствующая дизайну программного обеспечения и приоритетным требованиям к программному обеспечению. | | |
| 6.4.8.2.b | Определены подходы и контрольные точки для корректной работы собранных интерфейсов и функций системы. | | | 7.1.6.2.a,  7.1.6.2.c | | Для модулей программного обеспечения разрабатывается стратегия интеграции, соответствующая дизайну программного обеспечения и приоритетным требованиям к программному обеспечению.  Элементы программного обеспечения проверяются с использованием определенных критериев. | | |
| 6.4.8.2.c | Доступны любые вспомогательные системы или услуги, необходимые для интеграции. | | |  | | | | |
| 6.4.8.2.d | Интегрируется система, состоящая из реализованных системных элементов. | | | 7.1.6.2.d | | Производятся элементы программного обеспечения, определенные стратегией интеграции. | | |
| 6.4.8.2.e | Проверяются интерфейсы между реализованными элементами системы, составляющими систему. | | | 7.1.6.2.c | | Элементы программного обеспечения проверяются с использованием определенных критериев. | | |
| 6.4.8.2.f | Проверяются интерфейсы между системой и внешней средой. | | | 7.1.6.2.c | | Элементы программного обеспечения проверяются с использованием определенных критериев. | | |
| 6.4.8.2.g | Выявлены результаты интеграции и аномалии. | | | 7.1.6.2.c,  7.1.6.2.e | | Элементы программного обеспечения проверяются с использованием определенных критериев.  Результаты интеграционного тестирования фиксируются. | | |
| 6.4.8.2.h | Установлена прослеживаемость элементов интегрированной системы. | | | 7.1.6.2.f | | Между разработкой программного обеспечения и элементами программного обеспечения устанавливаются согласованность и прослеживаемость. | | |
| 6.4.9 Процесс верификации | | | | 7.1.5 Процесс создания программного обеспечения | | | | |
| 6.4.9.2.a | Выявлены ограничения проверки, влияющие на требования, архитектуру или дизайн. | | | 7.1.5.2.a | | Критерии проверки определены для всех программных модулей на соответствие их требованиям. | | |
|  | | | | 7.1.6 Процесс квалификационного тестирования программного обеспечения | | | | |
| 6.4.9.2.a | Выявлены ограничения проверки, влияющие на требования, архитектуру или дизайн. | | | 7.1.6.2.b,  7.1.6.2.g | | Для элементов программного обеспечения разработаны критерии проверки, обеспечивающие соответствие требованиям к программному обеспечению, назначенным элементам.  Стратегия регрессии разрабатывается и применяется для повторной проверки программных элементов, когда происходят изменения в программных единицах (включая связанные требования, дизайн и код). | | |
| 6.4.9.2.c | Система или системный элемент проверены. | | | 7.1.6.2.c | | Элементы программного обеспечения проверяются с использованием определенных критериев. | | |
|  | | | | 7.1.7 Процесс квалификационного тестирования программного обеспечения | | | | |
| 6.4.9.2.a | | Выявлены ограничения проверки, влияющие на требования, архитектуру или дизайн. | | 7.1.7.2.a,  7.1.7.2.d | | | Разработаны критерии для интегрированного программного обеспечения, демонстрирующие соответствие программным требованиям.  Стратегия регрессии разрабатывается и применяется для повторного тестирования интегрированного программного обеспечения при внесении изменений в элементы программного обеспечения. | |
| 6.4.9.2.c | | Система или системный элемент проверены. | | 7.1.7.2.b,  7.1.7.2.d | | | Интегрированное программное обеспечение проверяется с использованием определенных критериев.  Стратегия регрессии разрабатывается и применяется для повторного тестирования интегрированного программного обеспечения при внесении изменений в элементы программного обеспечения. | |
| 6.4.9.2.e | | Предоставляется объективное свидетельство того, что реализованная система соответствует требованиям, архитектуре и дизайну. | | 7.1.7.2.c | | | Результаты испытаний записываются. | |
| 6.4.9.2.f | | Выявлены результаты поверки и аномалии. | | 7.1.7.2.c | | | Результаты испытаний записываются. | |
| 6.4.9.2.g | | Установлена прослеживаемость проверенных элементов системы. | | 7.1.7.2.a | | | Разработаны критерии для интегрированного программного обеспечения, демонстрирующие соответствие программным требованиям. | |
|  | | | | 7.2.3 Процесс обеспечения качества программного обеспечения | | | | |
| 6.4.9.2.c | | Система или системный элемент проверены. | | 7.2.3.2.d | | | Проверяется соответствие программных продуктов, процессов и действий применимым стандартам, процедурам и требованиям. | |
|  | | | | 7.2.4 Процесс верификации программного обеспечения | | | | |
| 6.4.9.2.a | | Выявлены ограничения проверки, влияющие на требования, архитектуру или дизайн. | | 7.2.4.2.a,  7.2.4.2.b | | | Разработана и внедрена стратегия проверки программного обеспечения.  Определены критерии проверки всех требуемых программных продуктов. | |
| 6.4.9.2.b | | Доступны любые вспомогательные системы или услуги, необходимые для проверки. | | 7.2.4.2.a | | | Разработана и внедрена стратегия проверки программного обеспечения. | |
| 6.4.9.2.c | | Система или системный элемент проверены. | | 7.2.4.2.c | | | Необходимые действия по проверке выполняются. | |
| 6.4.9.2.d | | Приведены данные, предоставляющие информацию для корректирующих действий. | | 7.2.4.2.d | | | Дефекты выявляются и регистрируются. | |
| 6.4.9.2.e | | Предоставляется объективное свидетельство того, что реализованная система соответствует требованиям, архитектуре и дизайну. | | 7.2.4.2.e | | | Результаты проверки предоставляются заказчику и другим вовлеченным сторонам. | |
| 6.4.9.2.f | | Выявлены результаты поверки и ошибки. | | 7.2.4.2.d | | | Дефекты выявляются и регистрируются. | |
| 6.4.9.2.g | | Установлена прослеживаемость проверенных элементов системы. | | 7.2.4.2.b,  7.2.4.2.c | | | Определены критерии проверки всех требуемых программных продуктов.  Необходимые действия по проверке выполняются. | |
| 6.4.10 | | Переходный процесс | | 6.2.4 Процесс управления человеческими ресурсами | | | | |
| 6.4.10.2.d | | Операторы, пользователи и другие заинтересованные стороны, необходимые для использования и поддержки системы, проходят обучение. | | 6.2.4.2.c | | | Навыки персонала развиваются, поддерживаются или повышаются. | |
|  | | | | 6.4.7 Процесс установки программного обеспечения | | | | |
| 6.4.10.2.a | | Выявлены ограничения перехода, которые влияют на системные / программные требования, архитектуру или дизайн. | | 6.4.7.2.a,  6.4.7.2.b | | | Разработана стратегия установки программного обеспечения.  Разработаны критерии установки программного обеспечения, демонстрирующие соответствие требованиям к установке программного обеспечения. | |
| 6.4.10.2.b | | Доступны любые вспомогательные системы или услуги, необходимые для перехода. | | 6.4.7.2.d | | | Готовность программного продукта к использованию в предполагаемой среде гарантирована. | |
| 6.4.10.2.c | | Сайт подготовлен. | | 6.4.7.2.d | | | Готовность программного продукта к использованию в предполагаемой среде гарантирована. | |
| 6.4.10.2.d | | Система, установленная на рабочем месте, способна выполнять указанные функции. | | 6.4.7.2.d | | | Готовность программного продукта к использованию в предполагаемой среде гарантирована. | |
| 6.4.10.2.e | | Операторы, пользователи и другие заинтересованные стороны, необходимые для использования и поддержки системы, прошли обучение. | | 6.4.7.2.d | | | Готовность программного продукта к использованию в предполагаемой среде гарантирована. | |
| 6.4.10.2.f | | Выявлены результаты перехода и ошибки. | | 6.4.7.2.d | | | Готовность программного продукта к использованию в предполагаемой среде гарантирована. | |
| 6.4.10.2.g | | Установленная система активирована и готова к работе. | | 6.4.7.2.c,  6.4.7.2.d | | | Программный продукт установлен в целевой среде.  Готовность программного продукта к использованию в предполагаемой среде гарантирована. | |
| 6.4.10.2.h | | Установлена прослеживаемость переходных элементов. | | 6.4.7.2.b | | | Разработаны критерии установки программного обеспечения, демонстрирующие соответствие требованиям к установке программного обеспечения. | |
|  | | | | 6.4.8 Процесс поддержки приемки программного обеспечения | | | | |
| 6.4.10.2.d | | Система, установленная на рабочем месте, способна выполнять указанные функции. | | 6.4.8.2.a,  6.4.8.2.b | | | Товар укомплектован и доставлен покупателю.  Поддерживаются приемочные испытания и проверки со стороны приобретателя. | |
| 6.4.10.2.e | | Операторы, пользователи и другие заинтересованные стороны, необходимые для использования и поддержки системы, прошли обучение. | | 6.4.8.2.c | | | Продукт вводится в эксплуатацию в среде клиентов. | |
| 6.4.10.2.f | | Выявлены результаты перехода и аномалии. | | 6.4.8.2.d | | | Проблемы, обнаруженные во время приемки, выявляются и сообщаются лицам, ответственным за решение. | |
| 6.4.10.2.g | | Установленная система активирована и готова к работе. | | 6.4.8.2.c | | | Продукт вводится в эксплуатацию в среде клиентов. | |
| 6.4.11 | | Процесс валидации | | 6.4.8 Процесс поддержки приемки программного обеспечения | | | | |
| 6.4.11.2.b | | Подтверждена доступность услуг, требуемых заинтересованными сторонами. | | 6.4.8.2.a,  6.4.8.2.b,  6.4.8.2.c | | | Продукт завершен и передан покупателю.  Поддерживаются приемочные испытания и проверки со стороны приобретателя.  Продукт вводится в эксплуатацию в среде заказчика. | |
| 6.4.11.2.d | | Система или системный элемент проверены. | | 6.4.8.2.b | | | Поддерживаются приемочные испытания и проверки эквайера. | |
| 6.4.11.2.f | | Выявлены результаты валидации и ошибки. | | 6.4.8.2.d | | | Проблемы, обнаруженные во время приемки, выявляются и сообщаются лицам, ответственным за решение. | |
|  | | | | 6.4.9 Процесс эксплуатации программного обеспечения | | | | |
| 6.4.11.2.d | | Система или системный элемент проверены. | | 6.4.9.2.c | | | Программное обеспечение протестировано и определено для работы в предполагаемой среде. | |
|  | | | | 7.2.5 Процесс валидации программного обеспечения | | | | |
| 6.4.11.2.a | | Определены критерии проверки требований заинтересованных сторон. | | 7.2.5.2.a,  7.2.5.2.b | | | Разработана и внедрена стратегия валидации.  Определены критерии валидации всех необходимых рабочих продуктов. | |
| 6.4.11.2.b | | Подтверждена доступность услуг, требуемых заинтересованными сторонами. | | 7.2.5.2.c | | | Выполняются необходимые действия по валидации. | |
| 6.4.11.2.c | | Выявлены ограничения валидации, влияющие на требования, архитектуру или дизайн. | | 7.2.5.2.a,  7.2.5.2.b | | | Разработана и внедрена стратегия валидации.  Определены критерии валидации всех необходимых рабочих продуктов. | |
| 6.4.11.2.d | | Система или системный элемент проверены. | | 7.2.5.2.c | | | Выполняются необходимые действия по валидации. | |
| 6.4.11.2.e | | Доступны любые вспомогательные системы или услуги, необходимые для проверки. | | 7.2.5.2.a | | | Разработана и внедрена стратегия валидации. | |
| 6.4.11.2.f | | Выявлены результаты валидации и аномалии. | | 7.2.5.2.d | | | Проблемы выявляются и записываются. | |
| 6.4.11.2.g | | Предоставляется объективное свидетельство того, что реализованная система или системный элемент удовлетворяет потребности заинтересованных сторон. | | 7.2.5.2.e,  7.2.5.2.f | | | Предоставляются доказательства того, что разработанные программные рабочие продукты подходят для предполагаемого использования.  Результаты валидации предоставляются заказчику и другим вовлеченным сторонам. | |
| 6.4.11.2.h | | Установлена прослеживаемость проверенных элементов системы. | | 7.2.5.2.b | | | Определены критерии валидации всех необходимых рабочих продуктов. | |
| 6.4.12 | | Процесс эксплуатации | | 6.4.8 Процесс поддержки приемки программного обеспечения | | | | |
| 6.4.12.2.d | | Предоставляются услуги системных продуктов, отвечающие требованиям заинтересованных сторон. | | 6.4.8.2.c | | | Продукт вводится в эксплуатацию в среде клиентов. | |
|  | | | | 6.4.9 Процесс эксплуатации программного обеспечения | | | | |
| 6.4.12.2.a | | Идентифицируются операционные ограничения, которые влияют на системные / программные требования, архитектуру или дизайн. | | 6.4.9.2.a | | | Определяется стратегия работы. | |
| 6.4.12.2.b | | Доступны любые вспомогательные системы, услуги и материалы, необходимые для работы. | | 6.4.9.2.b | | | Выявляются и оцениваются условия для правильной работы программного обеспечения в предполагаемой среде. | |
| 6.4.12.2.e | | Производительность системного продукта во время работы контролируется. | | 6.4.9.2.b | | | Выявляются и оцениваются условия для правильной работы программного обеспечения в предполагаемой среде. | |
| 6.4.12.2.d | | Предоставляются услуги системных продуктов, отвечающие требованиям заинтересованных сторон. | | 6.4.9.2.d | | | Программное обеспечение работает в предполагаемой среде. | |
| 6.4.12.2.f | | Оказывается поддержка заказчику. | | 6.4.9.2.e | | | Заказчикам программного продукта оказывается помощь и консультация в соответствии с соглашением. | |
| 6.4.13 | | Процесс технического обслуживания | | 6.4.10 | | | Процесс сопровождения программного обеспечения | |
| 6.4.13.2.a | | Выявлены ограничения обслуживания, которые влияют на системные требования, архитектуру или дизайн. | | 6.4.10.2.a,  6.4.10.2.b | | | Стратегия сопровождения разрабатывается для управления модификацией и миграцией продуктов в соответствии со стратегией выпуска.  Выявлено влияние изменений существующей системы на организацию, операции или интерфейсы. | |
| 6.4.13.2.b | | Доступны любые вспомогательные системы или услуги, необходимые для обслуживания. | | 6.4.10.2.d,  6.4.10.2.e | | | Модифицированные продукты разрабатываются с сопутствующими испытаниями, которые демонстрируют, что требования не нарушены.  Обновления продукта переносятся в среду заказчика.  Затронутые системы и программное обеспечение документация обновляется по мере необходимости. | |
| 6.4.13.2.c | | Доступны заменяемые, отремонтированные или измененные элементы системы. | | 6.4.10.c,  6.4.10.d,  6.4.10.e,  6.4.10.f | | | Модифицированные продукты разрабатываются с сопутствующими испытаниями, которые демонстрируют, что требования не нарушены.  Модификации продуктов переносятся в среду заказчика.  Модификация системного программного обеспечения доводится до сведения всех заинтересованных сторон. | |
| 6.4.13.2.d | | Сообщается о необходимости изменений для корректирующего, улучшающего или адаптивного обслуживания. | | 6.4.10.2.b | | | Выявлено влияние изменений существующей системы на организацию, операции или интерфейсы. | |
| 6.4.13.2.e | | Определяются данные о сбоях и сроке службы, включая сопутствующие расходы. | | 6.4.10.2.b | | | Выявлено влияние изменений существующей системы на организацию, операции или интерфейсы. | |
|  | | | | 7.3.1 Процесс разработки домена | | | | |
| 6.4.13.2.c | | Доступны заменяемые, отремонтированные или измененные элементы системы. | | 7.3.1.2.f | | | Активы, принадлежащие домену, приобретаются или развиваются и поддерживаются на протяжении всего их жизненного цикла. | |
| 6.4.14 | | Процесс утилизации | | 6.4.11 | | | Процесс утилизации программного обеспечения | |
| 6.4.14.2.a | | Ограничения по удалению предоставляются как входные данные для требований, архитектуры, дизайна и реализации. | | 6.4.11.2.a,  6.4.11.2.b | | | Определяется стратегия утилизации программного обеспечения. Ограничения по удалению представлены в качестве входных данных для требований. | |
| 6.4.14.2.b | | Доступны любые вспомогательные системы или услуги, необходимые для утилизации. | | 6.4.11.2.a,  6.4.11.2.c | | | Определяется стратегия утилизации программного обеспечения.  Элементы программного обеспечения системы уничтожаются или хранятся. | |
| 6.4.14.2.c | | Элементы системы или отходы уничтожаются, хранятся, утилизируются или перерабатываются в соответствии с требованиями, например требованиями безопасности. | | 6.4.11.2.c | | | Элементы программного обеспечения системы уничтожаются или хранятся. | |
| 6.4.14.2.d | | Среда возвращается в исходное или согласованное состояние. | | 6.4.11.2.d | | | Среда остается в согласованном состоянии. | |
| 6.4.14.2.e | | Доступны записи действий по утилизации и анализ. | | 6.4.11.2.e | | | Доступны записи, позволяющие сохранять знания о действиях по удалению и любой анализ долгосрочных воздействий. | |

# Библиография

|  |  |  |
| --- | --- | --- |
| [1] | ANSI/AIAA G ‐ 043A ‐ 2012e | Руководство ANSI/AIAA по подготовке рабочих концептуальных документов  (ANSI/AIAA Guide to the Preparation of Operational Concept Documents) |
| [2] | IEC 61508: 2010  (все части) | Функциональная безопасность электрических/электронных/программируемых электронных систем, связанных с безопасностью  (Functional safety of electrical/electronic/programmable electronic safetyrelated systems) |
| [3] | IEEE Std 1012™ – 2012 | Стандарт IEEE для проверки и валидации системы и программного обеспечения  (IEEE Standard for System and Software Verification and Validation) |
| [4] | IEEE Std 1062™ – 2015 | Рекомендуемая практика IEEE для приобретения программного обеспечения  (IEEE Recommended Practice for Software Acquisition) |
| [5] | IEEE Std 730™ – 2014 | Стандарт IEEE для процессов обеспечения качества программного обеспечения  (IEEE Standard for Software Quality Assurance Processes) |
| [6] | IEEE Std 828™ – 2012 | Стандарт IEEE для управления конфигурацией в системной и программной инженерии  (IEEE Standard for Configuration Management in Systems and Software Engineering) |
| [7] | INCOSE TP – 2003‐020‐01 | Технические измерения  (Technical Measurement) |
| [8] | INCOSE.2015 | Справочник по системной инженерии: Руководство по процессам и действиям жизненного цикла системы, версия  (Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities, version 4.0. Hoboken, NJ, USA: John Wiley and Sons, Inc., ISBN: 978‐1‐118‐99940‐0) |
| [9] | ISO 10004: 2012 | Менеджмент качества. Удовлетворенность потребителей. Руководящие указания по мониторингу и измерению  (Quality management – Customer satisfaction – Guidelines for monitoring and measuring) |
| [10] | ISO 10007: 2003 | Системы менеджмента качества. Руководящие указания по менеджменту конфигурации  (Quality management systems – Guidelines for configuration management) |
| [11] | ISO 14001: 2004 | Системы экологического менеджмента. Требования с руководством по применению  (Environmental management systems – Requirements with guidance for use) |
| [12] | ISO 15704: 2000 | Системы промышленной автоматизации. Требования к эталонным корпоративным архитектурам и методологиям  (Industrial automation systems – Requirements for enterprise-reference architectures and methodologies) |
| [13] | ISO 31000: 2009 | Менеджмент риска. Принципы и руководства  (Risk management – Principles and guidelines) |
| [14] | ISO 9000: 2015 | Системы менеджмента качества. Основы и словарь  (Quality management systems – Fundamentals and vocabulary) |
| [15] | ISO 9001: 2015 | Системы менеджмента качества. Требования  (Quality management systems – Requirements) |
| [16] | ISO 9004: 2009 | Системы менеджмента качества. Рекомендации по повышению производительности  (Quality management systems – Guidelines for performance improvements) |
| [17] | ISO 9241-210: 2010 | Эргономика взаимодействия человека и системы. Человеко-ориентированный дизайн для интерактивных систем  (Ergonomics of human-system interaction – Human-centred design for interactive systems) |
| [18] | ISO Guide 73: 2009 | Управление рисками. Словарь  (Risk management – Vocabulary) |
| [19] | ISO/FDIS 9241-220 | Эргономика взаимодействия человека и системы. Часть 220: Процессы для обеспечения, выполнения и оценки ориентированного на пользователя проектирования в организациях  (Ergonomics of human-system interaction – Part 220: Processes for enabling, executing and assessing human-centred design within organizations) |
| [20] | ISO TS 18152: 2010 | Эргономика взаимодействия человека и системы. Спецификация для оценки процессов, связанных с проблемами человека и системы  (Ergonomics of human-system interaction – Specification for the process assessment of human-system issues) |
| [21] | ISO/IEC 10746‐3: 2009 | Информационная технология. Открытая распределенная обработка. Эталонная модель: Архитектура  (Information technology – Open distributed processing – Reference model: Architecture) |
| [22] | ISO/IEC 15026-3: 2011 | Системная и программная инженерия. Гарантия систем и программного обеспечения. Часть 3. Уровни целостности системы  (System and software engineering – Systems and software assurance – Part 3: System integrity levels) |
| [23] | ISO/IEC 15026-4: 2012 | Системная и программная инженерия. Гарантия систем и программного обеспечения. Часть 4. Гарантия в жизненном цикле  (Systems and software engineering – Systems and software assurance – Part 4: Assurance in the life cycle) |
| [24] | ISO/IEC 15939: 2007 | Системная и программная инженерия. Процесс измерения  (Systems and software engineering – Measurement process) |
| [25] | ISO/IEC 16085: 2006 | Системная и программная инженерия. Управление жизненным циклом. Управление рисками  (System and Software Engineering – Life Cycle Management – Risk Management) |
| [26] | ISO/IEC 16350: 2015 | Информационные технологии. Системная и программная инженерия. Управление приложениями  (Information technology – Systems and software engineering – Application management) |
| [27] | ISO/IEC 19770‐1: 2012 | Информационные технологии. Управление программными активами. Часть 1. Процессы и многоуровневая оценка соответствия  (Information technology – Software asset management – Part 1: Processes and tiered assessment of conformance) |
| [28] | ISO/IEC 20000-1: 2011 (IEEE Std 20000-1: 2013) | Информационные технологии. Управление услугами. Часть 1: Требования к системе управления услугами  (Information technology – Service management – Part 1: Service management system requirements) |
| [29] | ISO/IEC 25010: 2011 | Системная и программная инженерия. Требования и оценка качества систем и программного обеспечения (SQuaRE). Модели качества систем и программного обеспечения  (Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – System and software quality models) |
| [30] | ISO/IEC 25030: 2007 | Программная инженерия. Требования к качеству и оценка программного продукта (SQuaRE). Требования к качеству  (Software engineering – Software product Quality Requirements and Evaluation (SQuaRE) – Quality requirements) |
| [31] | ISO/IEC 25063: 2014 | Системная и программная инженерия. Требования к качеству и оценка систем и программных продуктов (SQuaRE). Общий отраслевой формат (CIF) для удобства использования: контекст использования, описание  (Systems and software engineering – Systems and software product Quality Requirements and Evaluation (SQuaRE) – Common Industry Format (CIF) for usability: Context of use description) |
| [32] | ISO/IEC 26550: 2013 | Разработка программного обеспечения и систем. Эталонная модель для проектирования и управления линейкой продуктов  (Software and systems engineering – Reference model for product line engineering and management) |
| [33] | ISO/IEC 27000: 2016 | Технология информации. Методы безопасности. Системы управления безопасностью информации. Обзор и словарь  (Information technology – Security techniques – Information security management systems – Overview and vocabulary) |
| [34] | ISO/IEC 27036 (все части) | Информационные технологии. Техники безопасности. Информация безопасность для отношений с поставщиками  (Information technology – Security techniques – Information security for supplier relationships) |
| [35] | ISO/IEC 33001: 2015 | Информационная технология. Оценка процесса. Понятия и терминология  (Information technology – Process assessment – Concepts and terminology) |
| [36] | ISO/IEC 33002: 2015 | Информационная технология. Оценка процесса. Требования к выполнению оценки процесса  (Information technology – Process assessment – Requirement for performing process assessment) |
| [37] | ISO/IEC 33004: 2015 | Информационная технология. Оценка процесса. Требования к эталонным процессам, оценке процессов и моделям зрелости  (Information technology – Process assessment – Requirement for process reference, process assessment, and maturity models) |
| [38] | ISO/IEC 33020: 2015 | Информационная технология. Оценка процесса. Структура измерения процесса для оценки возможностей процесса  (Information technology – Process assessment – Process measurement framework for assessment of process capability) |
| [39] | ISO/IEC TR 19759: 2015 | Руководство по сводам знаний по программной инженерии  (Guide to the Software Engineering Body of Knowledge (SWEBOK) V3, IEEE Computer Society, 2014) |
| [40] | ISO/IEC TR 24748-2: 2011 | Системная и программная инженерия. Управление жизненным циклом. Часть 2: Руководство по применению ISO/IEC 15288 (Процессы жизненного цикла системы)  (Systems and software engineering – Life cycle management – Part 2: Guide to the application of ISO/IEC 15288 (System life cycle processes)) |
| [41] | ISO/IEC TR 24748-3: 2011 | Системная и программная инженерия - Управление жизненным циклом. Часть 3: Руководство по применению ISO/IEC 12207 (Процессы жизненного цикла программного обеспечения)  (Systems and software engineering – Life cycle management – Part 3: Guide to the application of ISO/IEC 12207 (Software life cycle processes)) |
| [42] | ISO/IEC TR 24774: 2010 | Системная и программная инженерия. Управление жизненным циклом. Руководство по описанию процессов  (Systems and software engineering – Life cycle management – Guidelines for process description) |
| [43] | ISO/IEC TR 25060: 2010 | Системная и программная инженерия. Требования к качеству и оценка систем и программных продуктов (SQuaRE). Общий отраслевой формат (CIF) для удобства использования: Общие рамки для информации, связанной с удобством использования.  (Systems and software engineering – Systems and software product Quality Requirements and Evaluation (SQuaRE) – Common Industry Format (CIF) for usability: General framework for usability-related information) |
| [44] | ISO/IEC TS 24748-1: 2016 | Системная и программная инженерия. Управление жизненным циклом. Часть 1. Руководство по управлению жизненным циклом  (Systems and software engineering – Life cycle management – Part 1: Guide for life cycle management) |
| [45] | ISO/IEC/IEEE 14764: 2006 | Разработка программного обеспечения. Процессы жизненного цикла программного обеспечения. Обслуживание  (Software Engineering – Software Life Cycle Processes – Maintenance) |
| [46] | ISO/IEC/IEEE 15288: 2015 | Системная и программная инженерия. Процессы жизненного цикла системы  (Systems and software engineering – System life cycle processes) |
| [47] | ISO/IEC/IEEE 15289: 2015 | Системная и программная инженерия. Содержание информационных продуктов жизненного цикла (документация)  (Systems and software engineering – Content of life-cycle information products (documentation)) |
| [48] | ISO/IEC/IEEE 16326: 2009 | Системная и программная инженерия. Процессы жизненного цикла. Управление проектами  (Systems and software engineering – Life cycle processes – Project management) |
| [49] | ISO/IEC/IEEE 24748-4: 2016 | Системная инженерия. Управление жизненным циклом. Часть 4: Применение и управление процессом системной инженерии  (Systems engineering – Life cycle management – Part 4: Application and management of the systems engineering process) |
| [50] | ISO/IEC/IEEE 24748-5 | Системная и программная инженерия. Управление жизненным циклом. Часть 5. Планирование разработки программного обеспечения  (Systems and software engineering – Life cycle management – Part 5: Software development planning) |
| [51] | ISO/IEC/IEEE 24765 | Системная и программная инженерия. Словарь  (Systems and software engineering – Vocabulary) |
| [52] | ISO/IEC/IEEE 26515: 2011 | Системная и программная инженерия: Разработка пользовательской документации в гибкой среде  (Systems and software engineering: Developing user documentation in an agile environment) |
| [53] | ISO/IEC/IEEE 26531: 2015 | Системная и программная инженерия. Управление контентом для документации по управлению жизненным циклом продукта, пользователями и услугами  (Systems and software engineering – Content management for product life-cycle, user, and service management documentation) |
| [54] | ISO/IEC/IEEE 29119 (все части) | Системная и программная инженерия. Тестирование программного обеспечения  (Systems and software engineering – Software testing) |
| [55] | ISO/IEC/IEEE 29148: 2011 | Системная и программная инженерия. Процессы жизненного цикла  (Systems and software engineering – Life cycle processes – Requirements engineering) |
| [56] | ISO/IEC/IEEE 42010: 2011 | Системная и программная инженерия. Описание архитектуры  (Systems and software engineering – Architecture description) |
| [57] | НАТО AEP – 67 | Инженерное обеспечение системного обеспечения в программах НАТО  (Engineering for System Assurance in NATO Programs) |
| [58] | Практический стандарт PMI для структурной декомпозиции работ. Второе издание | |
| (PMI Practice Standard for Work Breakdown Structures‐Second Edition) | |
| [59] | SAE ANSI/EIA 649B | Стандарт управления конфигурацией  (Configuration Management Standard) |

**Исполнители**

Зам.директора по проектированию –

Руководитель службы ГИПов ОАО «Гипросвязь» А.Л.Ривкинд

Ведущий инженер НИЛ СУС НИИЦ

ОАО «Гипросвязь» М.Ю.Данилов