

ФГБОУ ВО «Воронежский государственный  
технический университет»

Е.Н. Королев

## МЕТОДЫ СИСТЕМНОЙ ИНЖЕНЕРИИ

Утверждено Редакционно-издательским советом  
университета в качестве учебного пособия

Воронеж 2016

УДК 681.38+681.3

Королев Е.Н. Методы системной инженерии: учеб. пособие [Электронный ресурс]. – Электрон. текстовые и граф. данные (1,6 Мб) / Е.Н. Королев. - Воронеж: ФГБОУ ВО «Воронежский государственный технический университет», 2016. – 1 электрон. опт. диск (CD-ROM) : цв. – Систем. требования : ПК 500 и выше ; 256 Мб ОЗУ ; Windows XP ; SVGA с разрешением 1024x768 ; Adobe Acrobat; CD-ROM дисковод ; мышь. – Загл. с экрана.

Учебное пособие по курсу «Системная инженерия» предназначено для получения студентами целостного представления о системной инженерии как междисциплинарной области технических наук, сосредоточенной на проблемах разработки и построения сложных, комплексных искусственных систем.

Издание соответствует требованиям Федерального государственного образовательного стандарта 09.04.01 по направлению «Информатика и вычислительная техника», дисциплине «Системная инженерия».

Ил. 19. Библиогр.: 3 назв.

Рецензенты: учебно-методический совет Воронежского института высоких технологий  
(канд. техн. наук, доц. Ю.П. Преображенский);  
канд. техн. наук, проф. С.М. Пасмурнов

© Королев Е.Н., 2016

© Оформление. ФГБОУ ВО «Воронежский государственный технический университет», 2016

## ВВЕДЕНИЕ

Системная инженерия это междисциплинарный подход и средства для создания систем; междисциплинарный подход, охватывающий все технические усилия по развитию и верификации интегрированного и сбалансированного в жизненном цикле множества системных решений по созданию систем различных классов(программных и информационных), касающихся людей, продукта и процесса, которые удовлетворяют потребности заказчика.

Системная инженерия — это практическое применение научных, инженерных и управленческих навыков, необходимых для преобразования операционных требований в описание конфигурации системы, которая наилучшим образом удовлетворяет этим требованиям. Это общий процесс решения проблем, который применяется ко всему техническому управлению в проекте, посвященном разработке системы, предоставляя механизм формулирования и совершенствования определений изделий и процессов системы.

Системная инженерия включает в себя пять функций.

– Определение проблемы, то есть указание потребностей и ограничений путем анализа требований и взаимодействия с заказчиком.

– Анализ решений, то есть выделение набора возможных способов удовлетворения потребностей и ограничений, их анализ и выбор оптимального.

– Планирование процессов, то есть определение задач, которые должны быть выполнены, объема ресурсов и затрат, необходимых для создания изделия, очередности задач и потенциальных рисков.

– Контроль процессов, то есть определение методов мониторинга проекта и процессов, измерение прогресса, оценка промежуточных изделий и принятие по мере необходимости корректирующих действий.

– Оценка изделий, то есть определение качества и количества создаваемых изделий путем оценочного планирования, тестирования, демонстрации, анализа, верификации и контроля.

Системная инженерия на всех этапах от выявления нужд потребителей и до определения необходимых функциональных возможностей системы, от документирования требований и до разработки и реализации проекта и проверки соответствия сосредоточена на всестороннем рассмотрении полного ЖЦ системы, включая:

- Обеспечение заданных функциональных возможностей и характеристик,
- Соблюдение смет и графиков,
- Контроль соответствия,
- Производство и сопровождение,
- Обучение персонала,
- Вывод из эксплуатации и утилизацию.

Современная системная инженерия, используя широкий арсенал методов, средств и инструментов, включая различные виды моделирования, принятия решений и оптимизации, управления рисками, планирования и управления сложностью сосредотачивает усилия в первую очередь на двух основных аспектах создания систем:

- Системном взгляде на продукцию и услуги,
- Методах разработки с использованием базовых моделей и типовых процессов.

Целью дисциплины «Системная инженерия» является получение студентами целостного представления о системной инженерии, как междисциплинарной области технических наук, сосредоточенной на проблемах разработки и построения сложных, комплексных искусственных систем.

В процессе изучения дисциплины студенты получают теоретические знания о комплексе технических, организационных и управленческих вопросов создания эффективных си-

стем, а также практические навыки по разработке моделей процессов системной инженерии и жизненного цикла систем.

Для достижения указанной цели в процессе изучения дисциплины решаются следующие задачи:

1. овладение знаниями и достижение понимания:

- целей и задач системной инженерии, как комплексной дисциплины, обеспечивающей успешную реализацию коллективных усилий по формированию и осуществлению набора процессов, необходимых для построения системы в ее развитии;

- роли и места системного инженера в процессе создания сложных систем;

- основных системных концепций в их связи с положениями основополагающих стандартов в области системной и программной инженерии;

- целей, задач и организации работ по стандартизации в области системной и программной инженерии;

- назначения и рекомендаций по применению основных нормативных документов в области системной и программной инженерии, на примере официальных и фактических стандартов;

- характеристик и особенностей практического применения процессов жизненного цикла систем и программных средств на примере стандартов группы ИСО 15288 и ИСО 12207;

- проблемы принятия решений при создании сложных систем;

- современных подходов к реализации технических процессов жизненного цикла систем, в первую очередь, процесса проектирования архитектуры

2. получение умений и навыков:

- планирования жизненного цикла сложной системы;

- формирования набора моделей, необходимых для успешного создания программно-интенсивных систем;

- принятия решений при выборе компонентов, необходимых для создания системы.

По итогам освоения дисциплины учащийся должен овладеть следующими компетенциями:

1. Иметь представление:

- о новых методах и технологиях в области ИС;
- о системе международных и национальных стандартов в области системной и программной инженерии;
- об особенностях реализации процессов системной инженерии на примере конкретного проекта IBS;
- о перспективах развития ИТ и ИС в бизнесе и государстве;
- о лучших практиках реализации процессов системной и программной инженерии и тенденциях их развития;
- о технологии проектирования корпоративных систем управления с использованием программных инструментальных средств;
- о технических средствах и инструментах реализации процессов системной и программной инженерии.

2. Знать:

- принципы построения и анализа систем управления; технологии проведения системно-аналитического обследования корпоративных систем управления;
- основные понятия и концепции системной инженерии;
- основные принципы и понятия процессного подхода к управлению и анализу систем управления;
- базовые методы и средства системной и программной инженерии;
- ГОСТы и международные стандарты в области ИТ;
- структуру и содержание основополагающих стандартов системной и программной инженерии;
- ГОСТ 34.601-90 Автоматизированные системы; стадии создания;

- ГОСТ 34.602-89 Техническое задание на создание автоматизированной системы;
  - РД 50-34.698-90 Автоматизированные системы; требования к содержанию документов;
  - методы принятия решений;
  - подходы к принятию решений при создании систем и к проектированию архитектуры программно-интенсивных систем;
  - методы управления развитием ИС; основы проектирования ИС (модель жизненного цикла, технологии и методологии разработки);
  - основы проектирования ИС: модель жизненного цикла, технологии и методологии разработки;
  - порядок планирования и реализации модели жизненного цикла и моделей процессов системной и программной инженерии при создании систем;
  - методы и средства сбора и разработки требований и спецификаций к элементам ИС;
  - основы системного анализа.
3. Уметь:
- использовать специализированные методологии и средства моделирования ИС, данных, процессов;
  - определять назначение и технические характеристики системы с учетом цели ее создания;
  - сопоставлять назначение и технические характеристики системы с составом и функциональными возможностями ее компонентов

# 1. ПРОЦЕССЫ ЖИЗНЕННОГО ЦИКЛА СИСТЕМЫ

Процессы жизненного цикла системы подразделяются на четыре группы:

- процессы соглашения;
- процессы предприятия;
- процессы проекта;
- технические процессы.

Каждый процесс жизненного цикла при необходимости может быть начат в любой момент жизненного цикла, при этом нет определенного порядка в их использовании. Любой процесс может выполняться одновременно с любыми другими процессами жизненного цикла и может быть реализован на любом уровне иерархии структуры системы. Таким образом, в следующем ниже описании процессов порядок, в котором представлены используемые процессы и группы процессов, не подразумевает предшествования или последовательности их применения в течение жизненного цикла системы.

Структура процессов жизненного цикла системы:

## 1. Процессы соглашения

- процесс приобретения
- процесс поставки

## 2. Процессы предприятия

- процесс управления средой предприятия
- процесс управления инвестициями
- процесс управления жц системы
- процесс управления ресурсами
- процесс управления качеством

## 3. Процессы проекта

- процесс планирования проекта;
- процесс оценки проекта;
- процесс контроля проекта;
- процесс принятия решений;
- процесс управления рисками;
- процесс управления конфигурацией;



– процесс управления информацией

#### 4. Технические процессы

- процесс определения требований правообладателей;
- процесс анализа требований;
- процесс проектирования архитектуры;
- процесс реализации элементов системы;
- процесс комплексирования;
- процесс верификации;
- процесс передачи;
- процесс валидации;
- процесс функционирования;
- процесс технического обслуживания;
- процесс изъятия и списания.

### 1.1 Процессы соглашения

Рассмотрим требования к процессам соглашения с организационными подразделениями, являющимися внешними и внутренними по отношению к организации. Процессы соглашения состоят из:

- процесса приобретения, используемого организациями для приобретения продукции или получения услуг;
- процесса поставки, используемого организациями для поставок продукции или оказания услуг.

Данные процессы определяют действия, необходимые для достижения соглашения между двумя организациями. В результате осуществления процесса приобретения обеспечиваются условия для ведения дел с поставщиком продукции, используемой как действующей системой и службами ее поддержки, так и элементами системы, разрабатываемой в рамках проекта. В результате процесса поставки обеспечиваются условия для управления проектом, результатом которого является продукт или услуга, поставляемые приобретающей стороне.

### 1.1.1 Процесс приобретения

Цель процесса приобретения состоит в получении продукта или услуги в соответствии с требованиями приобретающей стороны. В результате успешного осуществления процесса приобретения:

- определяется стратегия приобретения;
- выбирается поставщик;
- устанавливается связь с поставщиком;
- объявляется обоснование для выбора поставщика;
- заключается соглашение о приобретении продукта или услуги в соответствии с определенными
  - критериями приемки;
  - принимается продукт или услуга, соответствующие соглашению;
- осуществляется оплата или другие согласованные расчеты.

Приобретающая сторона должна осуществлять следующие действия в соответствии с принятой в организации политикой и процедурами в отношении процесса приобретения:

1. Утверждать план приобретения. Данный план включает ссылки на модель жизненного цикла, график контрольных сроков и критерий выбора подходящего поставщика, если поставщик является внешним по отношению к приобретающей организации.

2. Подготавливать заявку на поставку продукта или услуги. Необходимо обеспечить определение требований к одному или нескольким поставщикам. Если поставщик является внешним по отношению к организации, то заявка может включать правила деловых отношений, которым поставщик будет следовать, а также критерии выбора поставщика.

3. Передавать заявку на поставку продукции или услуг определенным поставщикам. К данному действию может относиться партнерство по управлению цепочкой поставок, которое позволяет обмениваться информацией со связан-

ными поставщиками и приобретающими сторонами для достижения, согласованного или коллективного подхода к общим техническим и коммерческим проблемам.

4. Выбирать поставщика. Для конкурсных поставок необходимо оценивать и сравнивать предложения по поставке на соответствие критериям выбора. Если предложения выходят за рамки критериев, они сравниваются друг с другом с целью определения их приоритетности и, как следствие, — предпочтительных поставщиков. Обоснование ранжирования каждого предложения документируется, и поставщики могут быть проинформированы о том, почему они были или не были выбраны.

5. Заключать соглашение с поставщиком. Данное соглашение может заключаться в различных формах — от письменного контракта до устного соглашения. В соответствии с принятой формой соглашения в нем устанавливаются требования к системным продуктам или услугам, контрольные сроки разработок и поставок, условия верификации, валидации и приемки, процедуры обработки исключительных ситуаций, процедуры контроля изменений и графики выплат. Таким образом, обе стороны соглашения вырабатывают условия для его выполнения. В соглашении указываются права и ограничения, связанные с техническими данными и интеллектуальной собственностью. Согласование является завершённым, когда приобретающая сторона принимает условия соглашения, предложенные поставщиком.

6. Оценивать выполнение соглашения. Это действие предполагает подтверждение того, что обе стороны выполняют свои обязательства в соответствии с соглашением. Планируемые технические риски, риски по срокам поставки и стоимостные риски должны контролироваться, а влияние на организацию нежелательных результатов регулярно оцениваться. При необходимости согласовывают изменения условий соглашения.

7. Подтверждать, что поставленный продукт или услуга соответствуют условиям соглашения. Отклонения,

возникающие при выполнении соглашения или в результате поставок продукта или услуги, разрешаются в соответствии с процедурами, установленными в соглашении.

8. Осуществлять оплату или обеспечивать другие согласованные расчеты с поставщиком продукта или оказанной услуги. Если поставленный продукт или услуга удовлетворяют условиям соглашения, приобретающая сторона завершает действие соглашения путем оплаты или других согласованных расчетов.

### **1.1.2 Процесс поставки**

Цель процесса поставки заключается в обеспечении приобретающей стороны продукцией или услугами, удовлетворяющими согласованным требованиям. В результате успешного осуществления процесса поставки:

- определяется приобретающая сторона продукта или услуги;
- составляется ответ на заявку приобретающей стороны;
- заключается соглашение о поставке продукта или услуги в соответствии с определенными критериями приемки;
- обеспечивается связь с приобретающей стороной;
- в соответствии с согласованными процедурами и условиями поставок поставляется продукт или услуга, удовлетворяющие соглашению;
- в порядке, указанном в соглашении, передается ответственность за приобретенный продукт или услугу;
- производится оплата или осуществляются другие согласованные взаиморасчеты.

При реализации процесса поставки поставщик должен осуществлять следующие действия в соответствии с принятыми в организации политикой и процедурами:

1. Определять наличие и подлинность приобретающей стороны, которая нуждается в продукте или услуге или представляет сторону или стороны, имеющие такую потребность. Для продукта или услуги, разработанных для потребителей, посредник, например, маркетинговый отдел внутри организации поставщика, может представлять приобретающую сторону.

2. Оценивать заявку на поставку продукта или услуги, чтобы определить ее выполнимость и содержание ответа на нее.

3. Готовить предложение по удовлетворению ходатайства.

4. Заключать соглашение с приобретающей стороной. Данное соглашение может заключаться в различных формах — от письменного контракта до устной договоренности. Необходимо преодолеть различия там, где они возникают, между заявкой на приобретение или заявленными потребностями, с одной стороны, и возможностями поставщика, выраженными в ответе на заявку, с другой стороны. Поставщик подтверждает, что требования, сроки поставки и условия приемки выполнимы, процедуры обработки исключительных ситуаций и контроля изменений, а также график оплаты приемлемы, и что все перечисленное является достаточным основанием выполнения соглашения без излишних рисков.

5. Выполнять соглашение в соответствии с утвержденными поставщиком проектными планами и в соответствии с текстом соглашения. Поставщик может принять или согласиться использовать процессы приобретающей стороны.

6. Оценивать выполнение соглашения. Риски, относящиеся к зафиксированной в соглашении стоимости, эксплуатационным характеристикам и срокам, контролируются, а информация о них соответствующим образом сообщается приобретающей стороне. Оценивается воздействие нежелательных результатов на деятельность организации.

7. Поставлять продукт или услугу в соответствии с критериями соглашения.

8. Принимать и подтверждать получение оплаты или выполнение других согласованных способов расчета.

9. Передавать ответственность за продукт или услугу приобретающей стороне или другой стороне в порядке, предусмотренном соглашением.

## **1.2 Процессы предприятия**

Процессы предприятия управляют способностью организации приобретать и поставлять продукцию или услуги посредством запуска проектов, их поддержки и контроля. Процессы предприятия обеспечивают ресурсы и инфраструктуру, необходимые для осуществления проектов, и гарантируют достижение целей и исполнение обязательств организации по соглашениям. Эти процессы не рассматриваются в качестве исчерпывающей совокупности бизнес-процессов, которые делают возможным стратегическое управление деятельностью организации.

Процессы предприятия включают в себя:

- процесс управления средой предприятия;
- процесс управления инвестициями;
- процесс управления процессами жизненного цикла системы;
- процесс управления ресурсами;
- процесс управления качеством.

### **1.2.1 Процесс управления средой предприятия**

Цель процесса управления средой предприятия заключается в определении и проведении политики и процедур, необходимых для функционирования организации в соответствии с положениями настоящего стандарта. В результате успешного осуществления процесса управления средой предприятия:

- реализуется политика и процедуры стратегического управления жизненным циклом системы;
- определяется степень ответственности и объем полномочий при осуществлении управления жизненным циклом системы;
- реализуется политика усовершенствования процессов жизненного цикла системы.

При реализации процессов управления средой предприятия необходимо осуществлять следующие действия в соответствии с принятой организацией политикой и процедурами:

1. Устанавливать планы действий для каждой области деятельности. Необходимо учесть краткосрочные цели, способствующие достижению стратегических целей, а также проекты, которые будут осуществляться для достижения стратегических целей.

2. Подготавливать политику и процедуры управления жизненным циклом системы, необходимые для реализации требований данного стандарта, не противоречащие стратегическому плану предприятия и планам действий для каждой области деятельности. Фактический объем операций и степень детализации проекта по осуществлению жизненного цикла систем зависят от сложности работ, используемых технологий, опытности и квалификации.

3. Персонала, выполняющего работу. Политика и процедуры должны соответствовать требованиям проекта, включая управление рисками, управление качеством и управление ресурсами.

4. Определять, интегрировать и связывать между собой роли, степень ответственности и полномочия для облегчения реализации процессов жизненного цикла системы и стратегического управления жизненным циклом системы.

5. Определять деловые критерии, позволяющие контролировать развитие процессов жизненного цикла. Следует установить критерии принятия решений, относящиеся к началу и окончанию.

6. Каждой стадии жизненного цикла, а также к другим ключевым событиям.

7. Периодически пересматривать используемую при проектировании модель жизненного цикла системы. Следует постоянно подтверждать пригодность, адекватность и результативность моделей жизненного цикла, используемых в каждом проекте, и соответствующим образом совершенствовать их.

8. Согласовывать с проектами политику и процедуры, принятые на предприятии, с целью реализации требований настоящего стандарта.

### **1.2.2 Процесс управления инвестициями**

Цель процесса управления инвестициями состоит в запуске в производство и поддержке обоснованных и успешных проектов, способствующих достижению целей организации. Управление инвестициями заключается в адекватном инвестировании фондов и ресурсов организации и в определении полномочий, необходимых для осуществления отобранных проектов. В процессе управления инвестициями осуществляется постоянная оценка проектов с целью подтверждения их обоснованности или доработки до приемлемого уровня и продолжения инвестирования.

В результате успешного выполнения процесса управления инвестициями:

- квалифицируются и отбираются инвестиционные возможности или потребности;
- определяются и распределяются ресурсы и денежные средства;
- определяются полномочия и ответственность при управлении проектом;
- поддерживаются проекты, удовлетворяющие условиям соглашения, требованиям правообладателей и организации;



– переориентируются, приостанавливаются или прекращаются проекты, не удовлетворяющие условиям соглашения, требованиям правообладателей или организации.

При реализации процессов управления инвестициями организация должна осуществлять следующие действия в соответствии с принятой политикой и процедурами:

1. Находить новые возможности и формы развития бизнеса, заключать новые соглашения, соответствующие стратегическому плану предприятия и планам для каждого из направлений его деятельности. Следует расставлять приоритеты для проектов, которые должны быть начаты, и устанавливать критерии для определения выполнимости этих проектов.

2. Структурировать проекты, определять ответственность и полномочия участников.

3. Оценивать предполагаемые результаты осуществления проектов.

4. Распределять ресурсы для достижения целей проекта.

5. Выявлять все возможные взаимосвязи проекта с другими проектами. В контексте настоящего стандарта в понятие «роль» могут входить в различных сочетаниях: наименование должности физического лица (наименование организации), выполняемые функции, взаимодействие и взаимосвязи с правообладателями или другими заинтересованными сторонами. Это действие включает применение обеспечивающих систем и (или) общих системных элементов одновременно для двух и более проектов.

6. Устанавливать требования к проектной отчетности и периодически оценивать реализацию ключевых событий, от которых зависит выполнение проекта.

7. Санкционировать начало выполнения утвержденных проектных планов, включая технические планы.

8. Оценивать текущие проекты с целью подтверждения того, что:

- проекты продвигаются в направлении достижения поставленных целей;
- проекты ведутся согласно соответствующим директивам;
- проекты реализуются в соответствии с планами и процедурами жизненного цикла систем;
- проекты остаются жизнеспособными, что подтверждается, например, постоянной потребностью в услуге, практичным выполнением продукта и приемлемыми доходами от инвестиций;

9. Принять решение о продолжении реализации или доработке проектов для того, чтобы их реализация прошла успешно;

10. В случаях, оговоренных соглашением, принять меры по отмене или приостановке тех проектов, ущерб или риски по которым превышают доходность инвестиций.

### **1.2.3 Процесс управления процессами жизненного цикла системы**

Цель процесса управления процессами жизненного цикла системы заключается в гарантировании доступности эффективных процессов жизненного цикла для использования организацией. Данный процесс обеспечивает процессы жизненного цикла системы, которые согласованы с целями и политикой организации, определены, адаптированы и поддерживаются соответствующим образом для учета особенностей отдельных проектов и способны реализовываться с помощью эффективных проверенных методов и инструментальных средств.

В результате эффективного управления процессами жизненного цикла системы:

- определяются процессы жизненного цикла системы, которые будут использоваться организацией;

- определяется политика применения процессов жизненного цикла системы;
- определяется политика адаптации процессов жизненного цикла системы для удовлетворения потребностей отдельных проектов;
- определяются критерии оценки результатов применения процессов жизненного цикла системы;
- предпринимаются действия по совершенствованию способов определения и применения процессов жизненного цикла системы.

При реализации процессов управления процессами жизненного цикла системы организация должна осуществлять следующие действия в соответствии с принятой политикой и процедурами:

1. Устанавливать стандартные наборы процессов жизненного цикла систем для соответствующих стадий жизненного цикла системы.
2. Определять приемлемые политику и процедуры адаптации и требования к их утверждению.
3. Определять методы и инструментальные средства, которые поддерживают выполнение процессов жизненного цикла системы.
4. По возможности устанавливать показатели, которые позволяют определять характеристики выполненных стандартных процессов.
5. Контролировать выполнение процесса, сохранять и анализировать показатели процесса и определять тенденции по отношению к критериям предприятия.
6. Определять возможности для совершенствования стандартных процессов жизненного цикла систем.
7. Совершенствовать имеющиеся процессы, методы и инструментальные средства, используя найденные возможности.

#### **1.2.4 Процесс управления ресурсами**

Цель процесса управления ресурсами состоит в обеспечении проектов необходимыми ресурсами. В результате процесса определяются ресурсы, материалы и услуги, необходимые для обеспечения организации и целей проектов в течение их жизненного цикла. В ресурсы включают квалифицированный, обученный и опытный персонал, способный реализовывать процессы жизненного цикла. Процесс управления ресурсами гарантирует эффективную координацию и совместное использование ресурсов, информации и технологий.

В результате успешного выполнения процесса управления ресурсами:

- проекты обеспечиваются необходимыми ресурсами, материалами и обслуживанием;
- поддерживается или улучшается квалификация персонала;
- разрешаются конфликты, возникающие в результате одновременного осуществления нескольких проектов.

При реализации процессов управления ресурсами организация должна осуществлять следующие действия в соответствии с принятой политикой и процедурами:

1. Определять и обеспечивать поддержку инфраструктуры ресурсов, необходимой для выполнения организацией требований настоящего стандарта и осуществления поддержки проекта. Проектные планы и будущие потребности бизнеса способствуют прояснению необходимой инфраструктуры ресурсов. Физические факторы, например, оборудование, и эргономические факторы, такие как уровень шума и параметры окружающей рабочей среды, считаются определенными.
2. Получать ресурсы, за исключением персонала, необходимые для внедрения и осуществления проектов.
3. Проявлять заботу о персонале, занятом в осуществлении текущих проектов. К этим действиям относятся: отбор, обучение и удержание персонала, обладающего необходимым уровнем навыков и опыта для того, чтобы про-

ект был должным образом обеспечен кадрами; контроль уровня компетентности персонала в области процессов жизненного цикла систем; обучение и тренировки персонала для совершенствования навыков и обеспечения возможности карьерного роста; оценка работы персонала по таким свойствам, как профессионализм, мотивация, способность работать в команде, необходимость в переобучении, в переводе на другую должность или в другое подразделение организации.

4. Стимулировать персонал, например, посредством предоставления возможности карьерного роста или при помощи системы поощрений.

5. Контролировать области взаимодействия нескольких проектов для разрешения связанных с графиками их реализации конфликтов:

- из-за ограниченных возможностей организационной инфраструктуры, вспомогательных служб и ресурсов при распределении между текущими проектами;

- из-за занятости персонала работой над несколькими проектами одновременно.

### **1.2.5 Процесс управления качеством**

Цель процесса управления качеством состоит в том, чтобы обеспечить такой уровень качества продукции, услуг и реализации процессов жизненного цикла, который бы соответствовал целям предприятия в области качества и удовлетворял заказчика.

В результате успешного осуществления процесса управления качеством:

- определяются политика организации и процедуры в области управления качеством;

- определяются цели и задачи организации в области управления качеством;

- определяется ответственность и полномочия для управления качеством;

- контролируется степень удовлетворенности заказчика;
- предпринимаются необходимые меры в случае, если цели в области управления качеством не достигнуты.

При реализации процессов управления качеством организация должна осуществлять следующие действия в соответствии с принятыми политикой и процедурами:

1. Устанавливать политику, стандарты и процедуры управления качеством.
2. Устанавливать цели организации в области управления качеством, основанные на стратегии, направленной на обеспечение удовлетворенности заказчика.
3. Определять ответственность и полномочия при реализации управления качеством.
4. Проводить оценку и составлять отчеты о степени удовлетворенности заказчика. Выполнение настоящего стандарта предоставляет организации подход к достижению удовлетворенности заказчика.
5. Проводить периодическую переоценку планов обеспечения качества проектов. Необходимо убедиться, что цели в области качества, основанные на требованиях правообладателя, установлены для каждого проекта.
6. Непрерывно контролировать состояние совершенствования качества продукции и услуг.

### **1.3 Процессы проекта**

Процессы проекта используются для установления и выполнения планов, оценки фактических достижений и продвижений проекта в соответствии с планами и для контроля выполнения проекта вплоть до его завершения. Отдельные процессы проекта могут осуществляться в любой момент жизненного цикла и на любом уровне иерархии проектов как в соответствии с проектными планами, так и с учетом непредвиденных обстоятельств. Уровень точности и формализации, с

которой осуществляются процессы проекта, зависит от сложности самого проекта и проектных рисков.

Процессы проекта состоят из следующих процессов:

- процесс планирования проекта;
- процесс оценки проекта;
- процесс контроля проекта;
- процесс принятия решений;
- процесс управления рисками;
- процесс управления конфигурацией;
- процесс управления информацией.

Планирование, оценка и контроль являются ключевыми процессами практически для всех видов управления. Они присутствуют в управлении любыми предпринимаемыми действиями, начиная с уровня всей организации и заканчивая каким-либо одним процессом жизненного цикла и его действиями.

### **1.3.1 Процесс планирования проекта**

Цель процесса планирования проекта состоит в составлении и доведении до заинтересованных сторон эффективного и выполнимого плана проекта. Этот процесс определяет область управления проектом и техническими мероприятиями, определяет результаты процесса, проектные задачи и поставки, устанавливает графики выполнения задач проекта, включая критерии достижения результатов и ресурсы, необходимые для выполнения задач проекта.

В результате успешного выполнения процесса планирования проекта:

- обеспечивается доступ к проектным планам;
- определяются роли, ответственность и полномочия участников;
- формируется официальный запрос на ресурсы и услуги, необходимые для достижения целей проекта;

- определяются показатели для характеристик проекта;
- штат проекта ориентируется в соответствии с планами проекта.

При реализации процесса планирования проекта организация должна осуществлять следующие действия в соответствии с принятой политикой и процедурами:

1. Определять проектные цели и ограничения. Цели и ограничения проекта включают рабочие характеристики и иные аспекты качества, затраты, сроки выполнения и показатели удовлетворенности правообладателей. Каждая цель проекта определяется с той степенью детализации, которая позволяет выбирать, настраивать и реализовывать соответствующие процессы и действия.

2. Определять границы проекта в соответствии с приглашением. В проект включаются все виды деятельности, необходимые для достижения соответствия критериям принятия бизнес-решений и успешного завершения проекта. Проект может отвечать за одну или несколько стадий полного жизненного цикла системы. Планирование включает в себя соответствующие действия по поддержанию проектных планов, оценке и контролю проекта.

3. Устанавливать декомпозицию работ, основанную на развивающейся системной архитектуре. Каждый элемент системной архитектуры, процессы и виды деятельности описываются со степенью детализации, согласованной с идентифицированными рисками. Задачи, соответствующие рабочей декомпозиции структуры, группируются в проектные задачи согласно организационным обязанностям. В свою очередь, проектные задачи идентифицируют каждый разрабатываемый или производимый рабочий продукт и связанные с ним задачи.

4. Определять и поддерживать графики работ в рамках проекта, основываясь на целях проекта и оценках выполнимости работ. К этим действиям относится определение продолжительности, взаимосвязей, зависимостей и последовательно-



сти мероприятий проекта, достижение контрольных точек его выполнения, используемые ресурсы и регулярно проводимый анализ (ревизии), необходимые для своевременного завершения проекта.

5. Определять критерии достижения результатов проекта для схем принятия решений на стадиях жизненного цикла, сроков поставок и основных зависимостей от внешних входов или выходов. Интервалы времени между внутренними пересмотрами проекта определяют в соответствии с политикой организации в отношении таких вопросов, как бизнес и критичность системы, графики работ и технические риски.

6. Определять расходы на проект и планировать бюджет. Расходы зависят от проектных графиков, оценок объемов работ, затрат на инфраструктуру, приобретаемые комплектующие, получение услуг, а также от затрат на обеспечивающие системы и резервов бюджета для управления рисками.

7. Устанавливать структуру полномочий и ответственности за выполнение работ в рамках проекта. Данное действие включает определение проектной организации, комплектование штата, развитие навыков персонала и методов его работы в команде. Также сюда входит эффективное использование человеческих ресурсов и тех организационных функций, которые выполняются на всех стадиях жизненного цикла системы. В структуре полномочий указываются юридически ответственные роли и лица, например, для полномочий в рамках проекта, полномочий по безопасности, для получения сертификата или аккредитации.

8. Определять инфраструктуру и службы, необходимые для реализации проекта. Это действие включает определение необходимых возможностей инфраструктуры и служб, их готовность и распределение по задачам проекта. Кроме того, сюда включают средства, инструменты, коммуникации и активы информационных технологий, задают также требования к обеспечивающим системам для каждой стадии жизненного цикла в пределах области применения проекта.

9. Планировать приобретение материалов, покупных изделий и услуг обеспечивающих систем для выполнения проекта. По мере необходимости сюда включают: планирование заявок, выбор поставщика, условия принятия, администрирование контракта и его завершение. Для запланированных приобретений используют процессы соглашения.

10. Формировать и доводить план до заинтересованных сторон для технического управления проектом, включая соответствующие ревизии.

11. Определять проектные показатели, которые должны быть сформированы, и связанные с ними данные, которые должны быть собраны, подвергнуты валидации и анализу. К этому действию относится определение источников проектных данных, их получателей и назначение соответствующих сроков.

12. Составлять планы по обеспечению качества проекта. Данное действие включает определение и документирование целей проекта в области качества, реализация которых служит гарантией достижения целей, политики и процедур управления качеством предприятия. Планирование должно осуществляться в соответствии с или другими стандартами в области качества.

### **1.3.2 Процесс оценки проекта**

Цель процесса оценки проекта заключается в определении статуса проекта. В ходе этого процесса периодически или при возникновении важных событий проводится оценка развития проекта и достижений относительно требований, планов и целей бизнеса. В случае обнаружения существенных отклонений информация о результатах оценки сообщается заинтересованным сторонам для осуществления адекватных управляющих воздействий.

В результате успешного осуществления процесса оценки проекта:

- становятся доступными показатели или результаты оценки рабочих характеристик проекта;
- оценивается адекватность ролей, обязанностей и полномочий участников проекта;
- оценивается адекватность ресурсов и услуг, необходимых для реализации проекта;
- анализируются отклонения от планируемых значений показателей рабочих характеристик проекта;
- заинтересованные стороны информируются о статусе проекта.

При реализации процесса оценки проекта организация должна осуществлять следующие действия в соответствии с проводимой политикой и установленными процедурами:

1. Оценивать статус проекта относительно соответствующих проектных планов для определения отклонений в затратах, сроках и качестве.
2. Обеспечивать гарантии качества в соответствии с проектными планами.
3. Оценивать результативность структуры команды участников проекта, распределения ролей и ответственности.
4. Оценивать адекватность и готовность инфраструктуры, обеспечивающей выполнение проекта.
5. Оценивать развитие проекта, используя измеренные достижения и результаты выполнения проекта в промежуточных контрольных точках.
6. Проводить требуемые управленческий и технический анализы, аудит и проверки для определения готовности к переходу на следующую стадию жизненного цикла системы или на следующий этап осуществления проекта.
7. Отслеживать критические процессы и новые технологии.
8. Анализировать данные и показатели для выявления значимых отклонений или изменений по отношению к

запланированным показателям и давать соответствующие рекомендации для корректировки.

9. Обеспечивать периодическую отчетность о статусе проекта и обязательную отчетность об отклонениях в соответствии с соглашением и принятыми в организации политикой и процедурами.

### **1.3.3 Процесс контроля проекта**

Цель процесса контроля проекта заключается в организации исполнения плана проекта и обеспечении гарантий реализации проекта в соответствии с планами и графиками в пределах бюджета проекта и гарантий удовлетворения технических целей.

При необходимости этот процесс включает в себя изменение направлений деятельности в рамках проекта, устранение выявленных отклонений и изменений, связанных с управлением другими проектами или техническими процессами. Соответственно, переориентирование может включать в себя перепланирование.

В результате успешного осуществления процесса контроля проекта:

- определяются и совершаются корректирующие действия, если результаты проекта не соответствуют запланированным заданиям;
- инициируется перепланирование проекта, если цели проекта или ограничения изменились, или допущения, сделанные при планировании, оказались неверными;
- санкционируются действия по переходу от одного запланированного этапа или события к следующему (при условии успешной реализации предыдущего этапа или события);
- достигаются цели проекта.

При реализации процесса контроля проекта организация должна осуществлять следующие действия в соответствии с принятой политикой и процедурами:

1. Управлять проектными требованиями и изменениями требований в соответствии с проектными планами.
2. Инициировать корректирующие действия, необходимые для достижения целей и получения результатов выполнения задач проекта при отклонениях свыше допустимых или заранее определенных пределов.
3. Соответствующим образом инициировать превентивные действия для гарантии достижения целей и результатов проекта.
4. Инициировать действия по разрешению проблем для коррекции несоответствий.
5. Разворачивать во времени содержание, определение и соответствующую декомпозицию работ, которые должны быть выполнены в рамках проекта вследствие принятых решений о корректирующих действиях и оцененных изменений, которые эти действия вносят.
6. По просьбе приобретающей стороны или поставщика инициировать действия, связанные с изменениями предусмотренных договором затрат, сроков или качества.
7. Осуществлять действия по исправлению нарушенных условий поставки приобретаемой продукции и услуг посредством конструктивного взаимодействия с поставщиком.
8. Санкционировать, если это обосновано, переход к реализации следующего запланированного этапа или события проекта.

### **1.3.4 Процесс принятия решений**

Цель процесса принятия решений заключается в выборе из существующих альтернатив наиболее предпочтительного направления проектных действий.

Этот процесс является реакцией на возникающие в процессе жизненного цикла системы запросы о принятии решений, направленных на достижение заданных, желаемых или оптимальных результатов вне зависимости от характера или

источников таких запросов. Альтернативные действия анализируются и выбирается направление действий. Решения и их обоснование документируются для поддержки принятия решений в будущем.

В результате успешного осуществления процесса принятия решений:

- определяется стратегия принятия решений;
- определяются альтернативные направления действий;
- выбирается наиболее предпочтительное направление действий;
- принятое решение, его обоснование и допущения документируются и доводятся до сведения заинтересованных сторон.

При реализации процесса принятия решений организация должна осуществлять следующие действия в соответствии с принятой политикой и процедурами:

1. Определять стратегию принятия решений.
2. Привлекать заинтересованные стороны к принятию решений для использования их опыта и знаний.
3. Устанавливать обстоятельства и необходимость принятия решений.
4. Выбирать и объявлять стратегию принятия решений для каждой ситуации, в которой необходимо принимать решение. Определять желаемые результаты и критерии успешного разрешения проблемы.
5. Оценивать баланс последствий альтернативных действий, используя определенную стратегию принятия решений, с целью оптимизации или улучшения ситуации принятия решений.
6. Документировать, отслеживать, оценивать и сообщать о результатах принятия решения для подтверждения эффективности решения проблем, устранения отрицательных тенденций и получения возможных преимуществ.
7. Поддерживать записи о проблемах и возможностях их решения, а также размещать эти записи в соответ-

ствии с соглашениями или организационными процедурами таким образом, который позволяет проводить аудит и изучать полученный опыт.

### **1.3.5 Процесс управления рисками**

Цель процесса управления рисками заключается в снижении последствий отрицательного воздействия вероятных событий, которые могут явиться причиной изменений качества, затрат, сроков или ухудшения технических характеристик.

В ходе данного процесса проводится определение, оценка, обработка и мониторинг рисков, возникающих в течение полного жизненного цикла, а также вырабатывается реакция на каждый риск в терминах реализации соответствующих мер противодействия риску или его принятия.

В результате успешного осуществления процесса управления рисками:

- определяются и классифицируются риски;
- количественно оцениваются вероятности и последствия осуществления рисков;
- устанавливается стратегия реакции на каждый из рисков;
- определяется и объявляется статус риска;
- принимаются соответствующие меры в случае, если риск вышел за пределы приемлемых значений.

В процессе управления рисками организация должна осуществлять следующие действия в соответствии с принятой политикой и процедурами:

1. Утвердить систематический подход к определению рисков, их оценке и обработке.
2. Идентифицировать и определять риски.
3. Определять вероятности событий, связанных с рисками, используя установленные критерии.

4. Оценивать риски в терминах их возможных последствий, используя установленные критерии.
5. Определять градации рисков по их вероятности и последствиям;
6. Определять стратегии реакции на риски.
7. Определять значения допустимых границ для каждого идентифицированного риска.
8. Определять действия по обработке рисков в случае превышения ими допустимых границ.
9. Сообщать о мерах по обработке рисков и их статусе в соответствии с действующими соглашениями, политикой и процедурами.
10. Вести учет рисков в течение всего жизненного цикла.

### **1.3.6 Процесс управления конфигурацией**

Цель процесса управления конфигурацией состоит в установлении и поддержании целостности всех идентифицированных выходных результатов проекта или процесса обеспечения доступа к ним любой заинтересованной стороны.

В результате успешного осуществления процесса управления конфигурацией:

- определяется стратегия управления конфигурацией;
- определяются элементы, нуждающиеся в управлении конфигурацией;
- устанавливается базовая линия конфигурации;
- контролируются изменения элементов, нуждающихся в управлении конфигурацией;
- контролируется конфигурация выделенных элементов;
- становится доступным на протяжении всего жизненного цикла статус элементов конфигурации, на которые распространяется управление.



При реализации процесса управления конфигурацией организация должна осуществлять следующие действия в соответствии с принятой политикой и процедурами:

1. определять стратегию управления конфигурацией.
2. идентифицировать элементы, которые необходимо контролировать в процессе управления конфигурацией.
3. поддерживать информацию о конфигурации на приемлемом уровне целостности и защищенности.
4. гарантировать, что изменения базовой линии конфигурации соответствующим образом идентифицируются, записываются, оцениваются, утверждаются, проводятся и верифицируются.

### **1.3.7 Процесс управления информацией**

Цель процесса управления информацией состоит в своевременном предоставлении заинтересованным сторонам необходимой полной, достоверной и, если требуется, конфиденциальной информации в течение и, соответственно, после завершения жизненного цикла системы.

В рамках процесса управления информацией реализуются функции создания, сбора, преобразования, хранения, восстановления, распространения и размещения информации. Этот процесс управляет перечисленной информацией, включая техническую и проектную информацию, информацию предприятия и пользовательскую информацию, а также информацию, содержащуюся в соглашениях.

В результате успешного осуществления процесса управления информацией:

- определяется информация, подлежащая управлению;
- определяются формы представления информации;
- информация преобразуется и распределяется в соответствии с требованиями;

- документируется статус информации;
- информация является «свежей», полной и достоверной;
- информация становится доступной для уполномоченных сторон.

При реализации процесса управления информацией организация должна осуществлять следующие действия в соответствии с принятой политикой и процедурами:

1. Определять элементы информации, которые будут подлежать управлению в течение жизненного цикла системы и, согласно политике организации или законодательству, поддерживаться в течение определенного периода после завершения жизненного цикла.

2. Распределять полномочия и обязанности, относящиеся к зарождению, созданию, накоплению, архивированию и уничтожению элементов информации.

3. Определять права, обязанности и обязательства, касающиеся хранения, передачи и доступа к элементам информации.

4. Определять содержание, семантику, форматы и средства для представления, хранения, передачи и поиска информации.

5. Получать идентифицированные элементы информации.

6. Обслуживать элементы информации и хранящиеся записи этих элементов в соответствии с требованиями к целостности, защите и сохранению тайны.

7. Определять действия по сопровождению информации.

8. Находить и распределять информацию между определенными сторонами в соответствии с требованиями согласованных графиков или при определенных обстоятельствах.

9. Предоставлять официальную документацию в соответствии с требованиями.

10. Архивировать заданную информацию в соответствии с целями аудита и сохранения знаний.

11. Уничтожить ненужную, искаженную или неподдающуюся проверке информацию в соответствии с политикой организации, требованиями к защите информации и сохранению тайны.

## **1.4 Технические процессы**

Технические процессы используются для определения требований к системе, преобразования этих требований в эффективный продукт, позволяющий осуществлять, при необходимости, устойчивое воспроизводство этого продукта, использовать его для обеспечения требуемых услуг, поддерживать обеспечение этими услугами и удалять продукт, когда он изымается из обращения. Технические процессы определяют совокупность работ, которые позволяют в рамках задач предприятия и проекта оптимизировать прибыли и уменьшать риски, возникающие вследствие принятия технических решений и осуществления соответствующих действий. Эти работы обеспечивают условия для того, чтобы продукция и услуги были нужными и полезными, экономически выгодными, функциональными, надежными, пригодными к обслуживанию, производству и использованию и обладали другими качествами, необходимыми для того, чтобы удовлетворить требования как приобретающих организаций, так и организаций-поставщиков. Они также обеспечивают условия для того, чтобы продукция и услуги соответствовали ожиданиям или законодательным требованиям общества, включая требования к факторам здоровья, безопасности, защиты и экологии.

Технические процессы включают в себя:

- процесс определения требований правообладателей;
- процесс анализа требований;
- процесс проектирования архитектуры;
- процесс реализации элементов системы;

- процесс комплексирования;
- процесс верификации;
- процесс передачи;
- процесс валидации;
- процесс функционирования;
- процесс технического обслуживания;
- процесс изъятия и списания.

#### **1.4.1 Процесс определения требований правообладателей**

Цель процесса определения требований правообладателей состоит в выявлении требований к системе, выполнение которых может обеспечить функциональные возможности, необходимые пользователям системы и иным заинтересованным лицам в заданной эксплуатационной среде.

Процесс позволяет определить правообладателей или классы правообладателей, которые связаны с системой на протяжении всего жизненного цикла, а также их потребности и пожелания. В рамках процесса эти данные анализируются и преобразуются в общий набор требований правообладателей, описывающих ожидаемое поведение системы в процессе взаимодействия с эксплуатационной средой, и совокупность базовых показателей, проверка на соответствие которым является целью процесса валидации, позволяющего подтвердить, что система отвечает заявленным требованиям.

В результате успешного осуществления процесса определения требований правообладателей:

- задаются требуемые характеристики и условия использования функциональных возможностей системы;
- определяются ограничения для системных решений;
- достигается возможность текущего отслеживания связей между требованиями правообладателей и самими правообладателями, и их потребностями;

- описывается основа для определения системных требований;
- определяется основа для валидации соответствия функциональных возможностей системы;
- формируется основа для ведения переговоров и заключения соглашения о поставке продукции или услуг.

При реализации процесса определения требований правообладателей организация должна осуществлять следующие действия в соответствии с принятой политикой и процедурами:

1. Идентифицировать отдельных правообладателей или классы правообладателей, имеющих законный интерес к системе в течение ее жизненного цикла.
2. Выявлять требования правообладателей.
3. Определять ограничения системных решений, которые являются неизбежным следствием существующих соглашений, управленческих или технических решений.
4. Определять представительный набор последовательных действий для идентификации всех требуемых функциональных возможностей, которые отвечают предполагаемым сценариям и средам функционирования и сопровождения.
5. Определять взаимодействие между пользователями и системой.
6. Устанавливать и специфицировать экологические, медицинские требования, требования безопасности и другие требования правообладателей, имеющие отношение к критическим показателям.
7. Анализировать полную совокупность выявленных требований.
8. Разрешать проблемы, возникающие в связи с определением требований.
9. Доводить результаты анализа требований до сведения соответствующих правообладателей для гарантии того, что их потребности и ожидания были правильно поняты и выражены.

10. Устанавливать совместно с правообладателями корректность выражения их требований.

11. Документировать требования правообладателей в форме, приемлемой для управления требованиями в течение жизненного цикла и за его пределами.

12. Поддерживать взаимное соответствие между требованиями правообладателей и потребностями заинтересованных лиц.

#### **1.4.2 Процесс анализа требований**

Цель процесса анализа требований состоит в преобразовании требований правообладателя, выраженных в виде его представлений о желаемых функциональных возможностях, в техническое видение требуемого продукта, способного предоставить такие функциональные возможности. В ходе этого процесса создается представление о будущей системе, которая сможет удовлетворить требования правообладателей и, если позволят ограничения, не подразумевают какой-либо специфической реализации. В результате данного процесса задаются измеримые системные требования, зависящие от видения разработчика, в которых определяется, какими характеристиками должна обладать система и какими должны быть значения этих характеристик, чтобы удовлетворить требования правообладателей.

В результате успешного осуществления процесса анализа требований:

- устанавливаются требуемые характеристики, свойства, функциональные и эксплуатационные требования к техническим решениям;

- устанавливаются ограничения, влияющие на архитектурное проектирование системы, а также на средства по его реализации;

- достигается целостность и прослеживаемость системных требований к требованиям правообладателей;

– определяется основа для верификации системных требований.

При реализации процесса анализа требований организация должна осуществлять следующие действия в соответствии с принятой политикой и процедурами:

1. Определять функциональные границы системы в терминах ее поведения и свойств, которые должны быть обеспечены.

2. Определять каждую функцию, которую система должна выполнять, насколько хорошо система, включая операторов, должна выполнять эту функцию, условия, при которых система способна выполнять данную функцию и при которых система начинает и прекращает ее выполнение.

3. Определять необходимые ограничения по изготовлению системы и ее элементов, которые обусловлены требованиями правообладателей или неизбежными ограничениями, связанными с принятием решений.

4. Определять технические показатели и показатели качества при использовании, позволяющие оценивать технические достижения.

5. Устанавливать системные требования и функции, в соответствии с которыми определяются риски и критические параметры системы, связанные с такими свойствами, как здоровье, безопасность, защищенность, безотказность, готовность, а также со свойствами обеспечивающих систем.

6. Анализировать целостность системных требований для обеспечения уверенности в том, что каждое требование, пары требований или наборы требований обладают системной целостностью.

7. Демонстрировать связь между системными требованиями и требованиями правообладателей.

8. На протяжении всего жизненного цикла вести учет совокупности системных требований вместе с их обоснованиями, связанными решениями и допущениями.

### **1.4.3 Процесс проектирования архитектуры**

Цель процесса проектирования архитектуры состоит в синтезе решения, которое бы удовлетворяло системным требованиям. Этот процесс выделяет и устанавливает области решения, представленные в виде набора различных проблем управленческого, концептуального и, наконец, реализационного характера. В рамках процесса определяются и исследуются одна или несколько стратегий реализации системы со степенью детализации, соответствующей техническим и коммерческим требованиям и рискам. Исходя из этого выбирается решение о проектировании архитектуры. Оно определяется на основе требований к набору системных элементов, из которых конструируется система. Конкретные требования, формируемые в результате этого процесса, являются основой для проведения верификации реализованной системы и для разработки стратегий комплексирования и верификации.

В результате успешного осуществления процесса проектирования архитектуры:

- устанавливается порядок, в соответствии с которым выполняется проектирование архитектуры;
- задается реализуемый набор описаний системных элементов, которые удовлетворяют требованиям, предъявляемым к системе;
- включаются в решение по проектированию архитектуры требования к интерфейсу;
- устанавливается связь между проектированием архитектуры и системными требованиями;
- определяется основа для верификации системных элементов;
- устанавливается основа комплексирования системных элементов.

При реализации процесса проектирования архитектуры организация должна осуществлять следующие действия в соответствии с принятой политикой и процедурами:



1. Определять приемлемые проекты логической архитектуры.
2. Выполнять декомпозицию функций системы, определенных в процессе анализа требований, и поставить им в соответствие элементы архитектуры системы, сформировать производные требования, необходимые для такого сопоставления.
3. Анализировать итоговый проект архитектуры с целью установления проектных критериев для каждого элемента.
4. Определять, какие системные требования должны выполняться операторами.
5. Определять, доступны ли в готовом виде те элементы технического и программного обеспечения, которые удовлетворяют проектным и интерфейсным критериям.
6. Оценивать альтернативные проектные решения, моделируя их с той степенью детализации, которая позволяет сравнивать спецификации, выраженные в системных требованиях, с эксплуатационными характеристиками, стоимостными и временными показателями и рисками, выраженными в требованиях правообладателей.
7. Определять и документировать области взаимодействия между системными элементами и области взаимодействий на границе системы с внешними системами.
8. Задавать выбранные физические проектные решения в соответствии с порядком проектирования архитектуры в терминах проектных функций, характеристик эксплуатации, поведения, интерфейсов и неизбежных ограничений при реализации проекта.
9. Вести документальный учет информации по проектированию архитектуры.
10. Поддерживать взаимосвязь и взаимозависимость между архитектурой и системными требованиями.

#### **1.4.4 Процесс реализации элементов системы**

Цель процесса реализации элементов системы состоит в создании заданных (специфицированных) элементов системы. В ходе этого процесса происходит преобразование заданных поведенческих, интерфейсных и производственных ограничений в действия по реализации, в результате которых в соответствии со сложившимися правилами и технологией создается элемент системы. Системный элемент конструируется или адаптируется путем обработки материалов и (или) информации, соответствующих выбранной технологии реализации, и использования соответствующих технических приемов и дисциплин. Результатом процесса является элемент системы, удовлетворяющий как архитектурным решениям, что подтверждается при верификации, так и требованиям правообладателей, что подтверждается при валидации.

В результате успешного осуществления процесса реализации элементов системы:

- определяется стратегия реализации элементов системы;
- определяются технологические ограничения, связанные с конструкцией системного элемента;
- изготавливается системный элемент;
- системный элемент упаковывается и хранится в соответствии с соглашением о его поставке.

При осуществлении процесса реализации элементов системы организация должна выполнять следующие действия в соответствии с принятой политикой и процедурами:

1. Разрабатывать стратегию реализации элементов системы.
2. Определять ограничения, которые стратегия и технология реализации элементов системы налагают на проектные решения.
3. Реализовывать или адаптировать системные элементы, используя обеспечивающие системы и определенные материалы в соответствии с заданными процедурами изготовления для производства технических средств, создания программных средств и (или) для обучения операторов.

4. Вести регистрацию доказательств соответствия элементов системы соглашениям с поставщиками, законодательству и политике организации.

5. Упаковывать элемент системы и хранить его в соответствующем виде.

### **1.4.5 Процесс комплексирования**

Цель процесса комплексирования заключается в сборке системы согласно архитектурному проекту. В ходе этого процесса системные элементы комбинируются таким образом, чтобы сформировать конфигурацию всей системы или ее части и создать продукт в соответствии с заданными системными требованиями.

В результате успешного осуществления процесса комплексирования:

- определяется стратегия комплексирования системы;
- определяются неизбежные ограничения, связанные с процессом комплексирования, которые влияют на системные требования;
- компонуется и комплексируется система, допускающая верификацию на соответствие требованиям, заданным архитектурным проектом;
- ведется документальный учет несоответствий, возникших в процессе комплексирования.

При реализации процесса комплексирования организация должна осуществлять следующие действия в соответствии с принятой политикой и процедурами:

1. Определять последовательность и стратегию сборки системы, которые минимизируют риски в процессе комплексирования. Такая стратегия позволяет осуществлять верификацию при помощи последовательности наращиваемых конфигураций системных элементов. Она зависит от степени готовности системных элементов и согласуется с локализацией последствий ошибок и стратегией оценки ситуации.

По возможности скомплексированная конфигурация включает в свой состав персонал операторов. Последовательное применение процессов комплексирования и верификации, а в случае необходимости и процесса валидации, повторяется для систем на каждом из последующих уровней до тех пор, пока рассматриваемая система не будет создана.

2. Идентифицировать ограничения на конструктивные решения, возникающие в результате следования стратегии комплексирования. К данному действию относится учет таких факторов, как доступность и комплексирование обеспечивающих систем, а также требуемые области сопряжения и взаимосвязи для промежуточных сборочных конфигураций.

3. Получать системы, обеспечивающие комплексирование, и необходимые материалы в соответствии с установленными процедурами комплексирования.

4. Получать системные элементы согласно графикам. Системные элементы могут быть получены от поставщиков или взяты из запасов. Обращение с системными элементами должно проводиться в соответствии с основными требованиями, связанными с обеспечением здоровья, безопасности, защиты и сохранения тайны.

5. Гарантировать, что системные элементы были верифицированы на соответствие критериям приемки, указанным в соглашении.

6. Комплексировать системные элементы в соответствии с применяемыми описаниями контроля интерфейсов и установленными процедурами сборки, используя заданные средства интеграции.

7. Вести учет информации, касающейся комплексирования, в соответствующей базе данных.

#### **1.4.6 Процесс верификации**

Цель процесса верификации состоит в подтверждении того, что заданные (специфицированные) требования проекта полностью реализованы в системе.

В ходе этого процесса получают информацию, которая требуется для совершения действий по устранению недостатков, что позволяет корректировать несоответствия в реализованной системе или процессы, происходящие в ней.

В результате успешного осуществления процесса верификации:

- определяется стратегия верификации;
- в качестве входных данных используются ограничения, накладываемые на верификацию;
- получаются отчетные данные, являющиеся источником для совершения корректирующих действий;
- предоставляются объективные доказательства того, что реализованная продукция удовлетворяет системным требованиям и требованиям архитектурного проекта.

При реализации процесса верификации организация должна осуществлять следующие действия в соответствии с принятой политикой и процедурами:

1. Определять стратегию верификации систем в течение жизненного цикла. Эта стратегия касается системы и ее описаний, например, требований, проектных определений. Она включает содержание и цели для каждого объекта верификации, например, при верификации проекта проверяется способность корректно осуществлять проектирование, способность к воспроизведению системы, возможность корректировать возникающие ошибки, способность прогнозировать отказы. Верификация демонстрирует посредством оценки продукта, что система создана «правильно», то есть система является реализацией того проекта, по которому и должен быть создан продукт. В ходе верификации, если есть возможность, в систему включается человек-оператор. Содержание и масштаб процесса верификации, например, пересмотр, инспекция, аудит, сравнение, статические испытания, динамические испытания, демонстрация (или комбинация этих видов верификации) зависят от того, что подвергается верификации: модель, прототип или реальный продукт, а также от возмож-

ных рисков, например, по безопасности, критичности с коммерческой точки зрения.

2. Определять план верификации, основываясь на системных требованиях. В планах учитывается последовательность конфигураций, определенных стратегией комплексования, и, если возможно, принимается в расчет стратегия демонтажа для диагностики ошибок. Графики обычно определяют этапы верификации, касающиеся управления рисками, которые последовательно обеспечивают уверенность в том, что продукт в максимальной конфигурации соответствует техническим условиям.

3. Идентифицировать и сообщать о потенциальных ограничениях на проектные решения.

4. Подготавливать обеспечивающую систему, а также соответствующие средства, оборудование и операторов к проведению верификации.

5. Осуществлять верификацию для демонстрации соответствия заданным проектным требованиям.

6. Формировать доступные верификационные данные о системе.

7. Анализировать и регистрировать информацию о верификации, отклонениях и корректирующих действиях, а также составлять соответствующие отчеты.

### **1.4.7 Процесс передачи**

Цель процесса передачи состоит в достижении способности обеспечивать услуги в среде функционирования согласно заданным требованиям правообладателей. В ходе этого процесса в соответствии с соглашениями приводится в рабочее состояние верифицированная система вместе с соответствующими обеспечивающими системами, например, операционной системой, системой поддержки, системой обучения операторов, системой обучения пользователей.

В результате успешного осуществления процесса передачи:

- определяется стратегия передачи;
- система приводится в рабочее состояние на месте ее применения;
- в процессе работы система способна выполнять свои функции;
- конфигурация приведенной в рабочее состояние системы документируется;
- регистрируются отчеты о корректирующих действиях;
- обеспечивающими системами предоставляются необходимые услуги.

В процессе передачи организация должна осуществлять следующие действия в соответствии с принятой политикой и процедурами:

1. Определять стратегию передачи.
2. Проводить подготовку места для размещения в соответствии с требованиями по установке.
3. Выполнить поставку системы в заданное место и в установленные сроки для приведения ее в рабочее состояние.
4. Установить систему на рабочем месте и связать ее со средой функционирования согласно спецификации.
5. Продемонстрировать, что система установлена надлежащим образом.
6. Активизировать систему;
7. Продемонстрировать способность установленной системы выполнять требуемые функции.
8. Вести документированный учет данных по установке, включая рабочую конфигурацию, обнаруженные отклонения, предпринятые действия и уроки, извлеченные из опыта этих действий.

#### **1.4.8 Процесс валидации**

Цель процесса валидации заключается в получении объективных доказательств того, что функции, обеспечиваемые системой при ее использовании, соответствуют требованиям правообладателей. В ходе данного процесса выполняется сравнительная оценка и подтверждается тот факт, что требования правообладателей правильно определены. В случае обнаружения отклонений они регистрируются и корректируются. Валидация системы утверждается правообладателями.

В результате успешного осуществления процесса валидации:

- определяется стратегия валидации;
- подтверждается готовность к выполнению функций, требуемых правообладателями;
- предоставляются данные валидации;
- составляется отчет по данным валидации, на основании которых можно осуществить корректирующие действия.

При реализации процесса валидации организация должна осуществлять следующие действия в соответствии с принятой политикой и процедурами:

1. Определять стратегию валидации реализуемых системой функций в среде функционирования при условии достижения удовлетворенности правообладателей.
2. Подготавливать план валидации.
3. Убеждаться в готовности операторов, обеспечивающих систем и соответствующего оборудования для проведения валидации.
4. Проводить валидацию для демонстрации соответствия функциональных возможностей системы требованиям правообладателей.
5. Приводить данные по валидации в соответствие с законодательством, регулируемыми требованиями или требованиями производственного сектора.
6. Согласно условиям соглашений или организационным целям проводить валидацию с изолированием той части системы, в которой могут возникать несоответствия.



7. Анализировать, регистрировать и составлять отчеты по валидационным данным в соответствии с критериями, определенными стратегией валидации.

При выполнении этого действия несоответствия классифицируются по их источнику и собственнику корректирующего действия. Валидационные данные анализируются для обнаружения таких важных признаков, как тенденции и условия отказов, доказательства ошибок проектирования и возникновения угроз функциональным возможностям системы.

### **1.4.9 Процесс функционирования**

Цель процесса функционирования состоит в использовании системы для выполнения заданных функций. В ходе этого процесса назначается персонал для работы в системе контроля выполнения функций и рабочих характеристик взаимодействия в звене «оператор—система». Для поддержания соответствующих услуг определяются и анализируются проблемы функционирования, связанные с соглашениями, требованиями правообладателей и организационными ограничениями.

В результате успешного осуществления процесса функционирования:

- определяется стратегия функционирования;
- поставляются услуги, удовлетворяющие требованиям правообладателей;
- успешно выполняются заявки на принятые корректирующие действия;
- поддерживается удовлетворенность правообладателей.

При реализации процесса функционирования организация в соответствии с принятой политикой и процедурами должна осуществлять следующие действия:

1. Подготавливать стратегию функционирования.
2. Получать другие услуги, относящиеся к функционированию системы.

3. Назначать на должности операторов обученный и квалифицированный персонал.

4. Активизировать систему в заданных условиях функционирования для представления примеров функций или продолжения непрерывного выполнения функций в соответствии с целевым назначением.

5. Применять материалы, требуемые для поддержания необходимых услуг.

6. Контролировать функционирование системы для подтверждения того, что система управляется в соответствии с планами работы, в безопасном режиме и в соответствии с законодательными актами, касающимися охраны труда и окружающей среды.

7. Осуществлять мониторинг функционирования системы для подтверждения того, что показатели выполнения функций находятся в пределах допустимых значений.

8. Осуществлять действия по обнаружению отказов при появлении несоответствий в выполняемых функциях.

9. Определять приемлемое направление действий, если требуется проведение корректирующих мероприятий для устранения ошибок, появившихся в результате изменений в потребностях.

10. Вводить необходимые изменения в порядок эксплуатации, среду функционирования, интерфейсы «человек-машина» и в обучение операторов, если ошибки человека приводят к отказам.

11. Постоянно или регулярно общаться с пользователями для определения степени, с которой предоставляемые услуги удовлетворяют их потребности.

#### **1.4.10 Процесс обслуживания**

Цель процесса обслуживания состоит в поддержании способности системы выполнять заданные функции. В ходе данного процесса контролируется способность системы выполнять заданные функции, регистрируются проблемы для

анализа, предпринимаются действия по корректировке, адаптации, исправлению и предупреждению нарушений функционирования, а также подтверждаются возможности выполнения функций в случае их восстановления после нарушений функционирования.

В результате успешного осуществления процесса обслуживания:

- разрабатывается стратегия обслуживания;
- ограничения процесса обслуживания применяются в качестве исходных данных при формировании системных требований;
- становятся доступными системные элементы, используемые для замены;
- осуществляется поддержка услуг, удовлетворяющих требования правообладателей;
- в отчетах сообщается о необходимости корректирующих проектных изменений;
- ведется документальный учет данных об отказах и сроках службы.

При реализации процесса обслуживания организация в соответствии с принятой политикой и процедурами должна осуществлять следующие действия:

1. Подготавливать стратегию обслуживания.
2. Определять ограничения системных требований, являющиеся неизбежным следствием реализации стратегии обслуживания.
3. Получать обеспечивающие системы, системные элементы и услуги, которые должны быть использованы при обслуживании системы.
4. Составлять отчеты о проблемах и вести записи об инцидентах с целью проведения диагностики отдельных событий и учета прошлых реализаций для поддержки последующего корректирующего, адаптирующего, совершенствующего или превентивного обслуживания.

5. Осуществлять процедуры исправления случайных неисправностей и (или) плановых замен системных элементов.

6. Инициировать корректирующие действия по устранению ранее необнаруженных конструкционных ошибок.

7. Подтверждать, что мероприятия по материально-техническому обеспечению удовлетворяют требуемым уровням пополнения запасов, в результате чего хранящиеся на складе системные элементы удовлетворяют требованиям по интенсивности восстановлений и запланированным срокам проведения технического обслуживания и ремонта.

8. Осуществлять превентивное обслуживание путем замены или обслуживания системных элементов до их отказа в соответствии с планами-графиками и процедурами технического обслуживания.

9. Выполнять действия по идентификации отказов при появлении любых несоответствий в системе.

10. Поддерживать составление отчетов, содержащих историю проблемы, выполненные корректирующие действия и обнаруженные тенденции для информирования операторов и персонала технического обслуживания и ремонта, а также лиц, занятых в других проектах, в которых создаются или используются подобные системные элементы.

#### **1.4.11 Процесс изъятия и списания**

Цель процесса изъятия и списания состоит в прекращении существования системного объекта. В течение этого процесса происходит деактивация, демонтаж и удаление системы и любых отходов, переход их в финальное состояние, возвращение окружающей среды к начальным или приемлемым условиям. В ходе данного процесса происходит уничтожение, сохранение или восстановление полезных свойств системного элемента и отходов экологически приемлемым способом в соответствии с законодательством, соглашениями, организаци-

онными ограничениями и требованиями правообладателей. При необходимости ведутся записи с целью контроля состояния здоровья операторов и пользователей, а также безопасности окружающей среды.

В результате успешного осуществления процесса изъятия и списания:

- определяется стратегия изъятия и списания;
- ограничения по изъятию и списанию предоставляются в качестве входных данных для требований;
- системные элементы уничтожаются, сохраняются, перерабатываются или восстанавливаются;
- окружающая среда возвращается к своему первоначальному или согласованному (с заинтересованными сторонами) состоянию;
- обеспечивается доступ к записям о действиях по изъятию и списанию и результатах анализа долгосрочных угроз.

При реализации процесса изъятия и списания организация в соответствии с принятой политикой и процедурами должна осуществлять следующие действия:

1. Определять стратегию изъятия и списания системы, включая каждый системный элемент и любые произведенные отходы.
2. Сообщать информацию о неизбежных ограничениях на конструкцию системы, вытекающих из стратегии изъятия и списания.
3. Приобретать обеспечивающие системы или получать услуги, которые будут использованы в процессе изъятия и списания системы.
4. Деактивировать систему с целью ее подготовки к удалению с места функционирования.
5. Выводить оперативный персонал из системы и выполнять записи полученных им оперативных знаний.
6. Расчленять систему на управляемые элементы с целью облегчения их удаления для повторного использования,

переработки, восстановления, переделки, архивирования или уничтожения.

7. Удалять систему из среды функционирования для повторного использования, переработки, восстановления, переделки или уничтожения.

8. Определять средства для хранения, места хранения, критерии для инспекций и периоды хранения, если система подлежит хранению.

9. При необходимости проводить уничтожение системы таким образом, чтобы понизить объемы обработки отходов или чтобы отходы было легче перерабатывать.

10. Подтверждать, что после изъятия и списания не существует вредных факторов для здоровья, безопасности, защищенности и окружающей среды.

11. Архивировать информацию, собранную в течение времени жизни системы, для проведения аудиторских проверок и анализа в случае, если существуют устойчивые угрозы здоровью, безопасности, защищенности и окружающей среде, а также для предоставления возможности последующим разработчикам и пользователям систем создавать базу знаний, используя накопленный опыт.

## **2. СТАДИИ ЖИЗНЕННОГО ЦИКЛА СИСТЕМЫ**

### **2.1 Стадия замысла**

Данная стадия начинается с момента осознания потребности или замысла создания новой или модификации существующей системы. Она является началом исследований, поиска фактов и периода планирования, когда оцениваются экономические, технические, стратегические и рыночные основы будущих действий через изучение приобретающей стороны и рынка, через анализ реализуемости и поиск компромиссов. Осуществляется обратная связь приобретающей стороны и пользователя с замыслом.

Одно или несколько альтернативных решений, удовлетворяющих идентифицированным потребностям или замыслу, разрабатываются посредством анализа, оценки реализуемости, выполнения приблизительных расчетов (таких как затраты, сроки, параметры рынка и логистики), изучения компромиссов, создания и демонстрации экспериментальных или опытных образцов. Определяется необходимость в одной или нескольких обеспечивающих системах для разработки, производства, применения, поддержки применения и списания рассматриваемой системы. В результате этой работы в оценку альтернатив включаются варианты решений с целью достижения сбалансированного решения по жизненному циклу системы. В качестве выходных результатов стадии являются требования правообладателей, концепции функционирования, оценки реализуемости, предварительные системные требования, примерные проектные решения, выраженные в форме чертежей, моделей, прототипов и т.п., а также планы стадии замысла для обеспечивающих систем, включая оценку стоимости всего времени их жизни, требований к человеческим ресурсам и предварительных проектных графиков. Принимается решение о продолжении выполнения работ на стадии разработки или об отказе от дальнейшей работы.

Предполагается, что у организации имеются в наличии методы, способы, инструментальные средства и компетентный персонал для проведения анализа рынка и экономического анализа, прогнозирования, анализа реализуемости проектных решений, анализа компромиссов, технического анализа, оценки общих затрат в течение времени жизни, моделирования, имитации или макетирования системы.

Стадия замысла выполняется для оценки новых возможностей в деловой сфере, разработки предварительных системных требований и осуществимых проектных решений.

Результаты выполнения стадии замысла перечислены ниже:

1. Установление новых замыслов, в которых предлагаются новые возможности, увеличение производительности

сти или снижение общей стоимости собственности правообладателей в течение жизненного цикла системы.

2. Оценка осуществимости замысла и решений для рассматриваемой системы в течение жизненного цикла, включая обеспечивающие системы, с учетом как технических, так и деловых целей правообладателя.

3. Подготовка и формирование базовой линии требований правообладателя и предварительных системных требований (технических спецификаций для выбранной рассматриваемой системы и пригодности спецификаций для предусмотренного способа взаимодействия между человеком и системой).

4. Уточнение результатов стадий в модели жизненного цикла системы.

5. Планы идентификации, оценки и уменьшения рисков для стадий модели жизненного цикла системы.

6. Идентификация и предварительная спецификация услуг, которые необходимо получать от обеспечивающих систем в течение жизненного цикла рассматриваемой системы.

7. Замыслы выполнения всех последующих стадий.

8. Планы и критерии завершения стадии разработки.

9. Планы идентификации, оценки и уменьшения рисков для данной и последующих стадий модели жизненного цикла системы.

10. Удовлетворение критерием завершения данной стадии.

11. Санкционирование перехода на стадию разработки.

## **2.2 Стадия разработки**

Стадия разработки начинается с достаточно детального технического уточнения системных требований и проектных



решений, их преобразования в один или несколько реализуемых продуктов, которые способны выполнять заданные функции в течение стадии использования по назначению. На этой стадии может использоваться прототип рассматриваемой системы. Соответствующим образом определяются, анализируются, проектируются, производятся, комплексируются, испытываются и оцениваются технические и программные средства и интерфейсы операторов, определяются требования к средствам производства, обучения и поддержки. На стадии разработки должны даваться гарантии того, что особенности последующих стадий (производство, применение, поддержка применения и списание), требований и возможностей обеспечивающих систем рассмотрены и учтены в проекте с привлечением всех заинтересованных сторон. Реализуется обратная связь между правообладателями и теми, кто будет производить, управлять, использовать, поддерживать и списывать рассматриваемую систему.

Результатом является рассматриваемая система или прототип рассматриваемой системы в ее окончательном виде, усовершенствованные обеспечивающие системы или имеющиеся обеспечивающие системы, вся документация и оценки стоимости последующих стадий.

Планирование на стадии разработки начинается на предыдущей стадии для гарантии того, что организация имеет в наличии или может создать инфраструктуру систем, обеспечивающих разработку, включающих методы, технические приемы, инструментальные средства и компетентный персонал, способный проводить анализ, моделирование и имитацию, макетирование, конструирование, комплексование, тестирование и разработку документации. Эти составляющие инфраструктуры разрабатываются или приобретаются для того, чтобы быть в наличии при необходимости поддерживать разработку.

Стадия разработки осуществляется с целью создания такой рассматриваемой системы, которая удовлетворяет требованиям приобретающей стороны и может быть создана, ис-

пытана, оценена, применена по назначению, поддержана при применении и списана.

Результатами стадии разработки являются:

1. Оцененные и уточненные системные требования, бюджет проекта, базовые сроки выполнения и оценки затрат для собственника жизненного цикла.

2. Архитектура рассматриваемой системы, состоящая из элементов программных и технических средств, людей и их интерфейсов (внутренних и внешних).

3. Документация по верификации и валидации.

4. Подтверждение того, что рассматриваемая система соответствует всем требованиям правообладателей и системным требованиям, может быть запущена в производство, применяться по назначению и поддерживаться в процессе применения, а также может переводиться в категорию непригодных к применению (списываться) и является эффективной по стоимости для правообладателей.

5. Уточненные и соответствующие базовой линии требования к обеспечивающим системам.

6. Техническая информация, в том числе:

- диаграммы, чертежи и модели технических средств;
- проектная программная документация;
- спецификации интерфейсов;
- производственные планы;
- рабочие инструкции;
- руководства по тренингу операторов;
- процедуры обслуживания;
- особенности изъятия и списания;

7. Прототип или непосредственно рассматриваемая система в окончательном виде.

8. Уточненные результаты и оценки затрат на стадиях производства, применения по назначению, поддержки применения, изъятия и списания.

9. Определения функциональных возможностей обеспечивающих систем, требуемых на последующих стадиях жизненного цикла.

10. Планы и критерии завершения стадии производства.

11. Идентифицированные текущие риски и определенные действия по их уменьшению.

12. Соответствие критериям перехода на следующую стадию.

13. Санкционирование перехода на стадию производства.

### **2.3 Стадия производства**

Данная стадия начинается с принятия к производству рассматриваемой системы. Рассматриваемая система может производиться, собираться, комплексоваться и испытываться в единственном экземпляре или может быть продуктом массового производства. Планирование для этой стадии начинается на предыдущей стадии

Производство может продолжаться на протяжении оставшегося периода жизненного цикла. В течение данной стадии продукт может быть улучшен или перепроектирован, обеспечивающие системы могут нуждаться в реконфигурации, а производственный персонал в переобучении для продолжения развития экономически эффективных, с точки зрения правообладателей, функциональных возможностей системы.

Предполагается, что в распоряжении организации имеется производственная инфраструктура, состоящая из бюджетных средств, производственного оборудования, инструментальных средств, процедур и компетентного персонала. Эти составляющие разрабатываются или приобретаются, чтобы быть в наличии при необходимости обеспечить производство.

Данная стадия может пересекаться со стадиями разработки, применения по назначению и поддержки в процессе применения.

Цель стадии производства состоит в производстве или изготовлении продукта, испытании продукта и производстве соответствующих необходимых поддерживающих и обеспечивающих систем.

Результаты стадии производства перечислены ниже:

- 1) оцениваются возможности производства;
- 2) приобретаются ресурсы, материалы, услуги и системные элементы для поддержки выполнения производственных заданий в количественном выражении;
- 3) производится продукт в соответствии с утвержденной и оцененной информацией о производстве;
- 4) упакованный продукт передается в каналы распределения или приобретающей стороне;
- 5) составляются планы и критерии завершения стадии применения и стадии поддержки применения;
- 6) формируются обновленные концепции осуществления всех последующих стадий;
- 7) определяются текущие риски и действия по их уменьшению;
- 8) рассматриваемая система принимается приобретающей стороной с гарантированным уровнем качества;
- 9) удовлетворяется критерий завершения стадии производства;
- 10) санкционируется переход на стадию применения.

## **2.4 Стадия применения**

Стадия применения начинается после установки и передачи системы для применения по назначению. Данная стадия осуществляется с целью использования продукта в предназначенном месте функционирования для предоставления требуемых услуг с продолжительной функциональной и стоимостной результативностью.

Эта стадия завершается, когда рассматриваемая система прекращает предоставление услуг.

Планирование для данной стадии начинается на предшествующих стадиях. Данная стадия включает процессы, относящиеся к использованию продукта для предоставления услуг, а также мониторинг характеристик функционирования, идентификацию, классификацию и составление отчетов об отклонениях, недостатках и отказах. Ответной реакцией на обнаруженные проблемы может быть отсутствие действия; техническое обслуживание и незначительная (с низкой стоимостью и кратковременная) модификация (относится к стадии поддержки применения; значительная (постоянная) модификация и продление срока жизни рассматриваемой системы (относится к стадии разработки и производства; изъятие и списание при окончании срока жизни (относится к стадии изъятия и списания).

В течение данной стадии продукт или услуги могут совершенствоваться, являясь, таким образом, причиной появления различных конфигураций. Пользователь оперирует различными конфигурациями, а ответственный поставщик продукции управляет статусом и описаниями различных версий и конфигураций используемых продуктов или услуг.

Предполагается, что организация имеет в своем распоряжении рабочую инфраструктуру, включающую устройства, оборудование, обученный персонал, эксплуатационную документацию и отлаженные процедуры. Эти составляющие разрабатываются или приобретаются для оперативной поддержки применения.

Стадия применения осуществляется для того, чтобы использовать продукт, предоставлять услуги в заданных условиях функционирования и гарантировать продолжительную результативность.

Результаты стадии применения перечислены ниже:

- 1) комплектуется опытный персонал с уровнем компетенции, необходимым для выполнения функций операторов в

рассматриваемой системе и предоставления соответствующих услуг;

2) на месте применения устанавливается рассматриваемая система, способная работать и предоставлять устойчивые функциональные услуги;

3) проводится мониторинг стоимости, рабочих характеристик и их оценка для подтверждения соответствия целям применения системы;

4) идентифицируются проблемы или недостатки, а соответствующие организации (пользователи, разработчики, производители или обслуживающие органы) информируются о необходимости проведения корректирующих действий;

5) выявляются и анализируются новые возможности для совершенствования рассматриваемой системы через обратную связь с правообладателем;

6) составляются планы и формируются критерии перехода на стадию изъятия и списания;

7) фиксируется удовлетворение критерия завершения данной стадии;

8) санкционируется переход на стадию изъятия и списания.

## **2.5 Стадия поддержки применения**

Стадия поддержки применения начинается с обеспечения техническим обслуживанием и сопровождением, материально-техническим снабжением и другими видами поддержки функционирования и использования рассматриваемой системы. Планирование для данной стадии начинается на предшествующих стадиях. Стадия поддержки применения завершается в момент прекращения применения и отмены поддерживающих услуг, в результате чего осуществляется переход на стадию изъятия и списания рассматриваемой системы.

Данная стадия включает процессы, относящиеся к функционированию поддерживающей системы и обеспече-

нию поддерживающих услуг пользователям рассматриваемой системы. Также на данной стадии осуществляется контроль рабочих характеристик служб и системы поддержки, идентификация, классификация и составление отчетов об аномалиях, недостатках и отказах служб и системы поддержки. Действия, которые предпринимаются в результате обнаружения проблем, включают техническое обслуживание и незначительные модификации служб и системы поддержки, существенные модификации служб или системы поддержки (относится к стадии разработки и производства, перевод служб и системы поддержки в категорию непригодных для использования в связи с истекшим сроком жизни (относится к стадии изъятия и списания).

В течение данной стадии службы, и система поддержки могут развиваться в виде различных версий или конфигураций. Организация, обеспечивающая поддержку, работает с этими версиями и конфигурациями, а организация, ответственная за продукт, осуществляет управление статусом и описаниями различных версий и конфигураций служб и системы поддержки при их функционировании.

Предполагается, что организация может обеспечивать поддержку, если она располагает участками для осуществления операций поддержки, устройствами, оборудованием и инструментами, обученным персоналом, инструкциями и процедурами по техническому обслуживанию и текущему ремонту. Составные части инфраструктуры поддержки разрабатываются и приобретаются для оперативной поддержки функционирования рассматриваемой системы.

Стадия поддержки применения осуществляется с целью осуществления материально-технического снабжения, технического обслуживания и текущего ремонта, которые обеспечивают непрерывное функционирование рассматриваемой системы и устойчивое предоставление услуг, поддерживающих ее применение.

Результаты стадии поддержки применения перечислены ниже:

1) комплектуется обученный персонал, который будет обслуживать и обеспечивать работу служб поддержки;

2) налаживаются организационные интерфейсы с техническими и производственными организациями, обеспечивающими гарантированное решение проблем и проведение корректирующих действий;

3) проводится обслуживание и предоставляются услуги, обеспечивающие все соответствующие службы поддержки (включая материально-техническое снабжение) на рабочих местах;

4) обеспечивается поддержка функционирования рассматриваемой системы, ее составных частей и предоставляемых ею услуг, исправляются недостатки, допущенные при проектировании;

5) обеспечивается поддержка всего необходимого материально-технического снабжения, в том числе запасными частями в количестве, которое требуется для достижения целей эксплуатационной готовности;

6) определяются текущие риски и предпринимаются действия для их уменьшения;

7) заключается соглашение об окончании функционирования служб поддержки;

8) устанавливается соответствие критерию завершения стадии.

## **2.6 Стадия изъятия и списания**

Стадия изъятия и списания обеспечивает ликвидацию рассматриваемой системы и связанных с нею эксплуатационных и поддерживающих служб. Планирование для стадии изъятия и списания начинается на предыдущих стадиях. Данная стадия начинается в момент снятия рассматриваемой системы с обслуживания.

Стадия изъятия и списания включает процессы, которые относятся к функционированию списываемой системы, а также включает мониторинг ее рабочих характеристик, опре-



деление, классификацию и составление отчетов об аномалиях, дефектах и отказах списываемой системы. Действия, предпринимаемые в результате обнаружения проблем, включают обслуживание и незначительные модификации списываемой системы (относятся к стадии поддержки применения (В.6), значительные модификации списываемой системы (относятся к стадии разработки и производства (В.3 и В.4), изъятие и списание системы по причине окончания срока жизни (относится к данной стадии).

Предполагается, что организация имеет доступ к инфраструктуре для поддержки изъятия и списания, включая средства, инструменты, оборудование и персонал, обученный соответствующим действиям и процедурам, и, при необходимости, доступ к средствам переработки, удаления или сохранения. Составляющие инфраструктуры списания разрабатываются или приобретаются для оперативного выполнения функции списания.

Данная стадия применяется, когда заканчивается срок выполнения функций рассматриваемой системы.

Окончание этого срока может наступить вследствие замещения новой системой, невозможности восстановления, катастрофического отказа, утраты интереса со стороны пользователя или в случае, когда продолжение применения и поддержки рассматриваемой системы экономически неэффективно.

Стадия изъятия и списания осуществляется с целью обеспечить удаление рассматриваемой системы и связанных с ней обслуживающих и поддерживающих служб из среды применения, непосредственно оперировать самой списываемой системой и поддерживать процесс ее изъятия и списания.

Результаты стадии изъятия и списания перечислены ниже:

- 1) комплектуется опытный персонал, способный обеспечить выполнение функций изъятия и списания;
- 2) прекращается применение рассматриваемой системы, включая ее удаление, обновление или переработку в соот-

ветствии с законодательством в области здравоохранения, безопасности, защиты, сохранения тайны и охраны окружающей среды;

3) формируются планы и процедуры передачи функций новой рассматриваемой системе (если это приемлемо);

4) удаляются отходы;

5) окружающая среда возвращается к первоначальному или согласованному состоянию;

6) обеспечивается архивирование элементов;

7) проводится перемещение, перевод или увольнение операторов;

8) констатируется удовлетворение критерия завершения стадии изъятия и списания.

### **3. ПОСТРОЕНИЕ ИНТЕГРИРОВАННОЙ МОДЕЛИ СЛОЖНОЙ СИСТЕМЫ**

#### **3.1. Язык моделирования UML**

Язык UML предоставляет в распоряжение пользователей легко воспринимаемый и выразительный язык визуального моделирования, специально предназначенный для разработки и документирования моделей сложных систем самого различного целевого назначения.

Конструктивное использование языка UML основывается на понимании общих принципов моделирования сложных систем и особенностей процесса объектно-ориентированного анализа и проектирования, в частности. Выбор выразительных средств для построения моделей сложных систем предопределяет те задачи, которые могут быть решены с использованием данных моделей. При этом одним из основных принципов построения моделей сложных систем является принцип абстрагирования, который предписывает включать в модель только те аспекты проектируемой системы, которые имеют непосредственное отношение к выполнению системой своих функций или своего целевого предназна-

чения. При этом все второстепенные детали опускаются, чтобы чрезмерно не усложнять процесс анализа и исследования полученной модели.

Другим принципом построения моделей сложных систем является принцип многомодельности. Этот принцип представляет собой утверждение о том, что никакая единственная модель не может с достаточной степенью адекватности описывать различные аспекты сложной системы. Применительно к методологии ООАП это означает, что достаточно полная модель сложной системы допускает некоторое число взаимосвязанных представлений (views), каждое из которых адекватно отражает некоторый аспект поведения или структуры системы. При этом наиболее общими представлениями сложной системы принято считать статическое и динамическое представления, которые в свою очередь могут подразделяться на другие более частные представления.)

Еще одним принципом прикладного системного анализа является принцип иерархического построения моделей сложных систем. Этот принцип предписывает рассматривать процесс построения модели на разных уровнях абстрагирования или детализации в рамках фиксированных представлений. При этом исходная или первоначальная модель сложной системы имеет наиболее общее представление (метапредставление). Такая модель строится на начальном этапе проектирования и может не содержать многих деталей и аспектов моделируемой системы.



Рис. 1. Общая схема взаимосвязей моделей и представлений сложной системы в процессе объектно-ориентированного анализа и проектирования

Формальное описание самого языка UML основывается на некоторой общей иерархической структуре модельных представлений, состоящей из четырех уровней:

- Мета-метамодель
- Метамодель
- Модель
- Объекты пользователя

Уровень мета-метамодели образует исходную основу для всех метамодельных представлений. Главное предназначение этого уровня состоит в том, чтобы определить язык для спецификации метамодели. Мета-метамодель определяет модель языка UML на самом высоком уровне абстракции и является наиболее компактным ее описанием. С другой стороны, мета-метамодель может специфицировать несколько метамodelей, чем достигается потенциальная гибкость включения дополнительных понятий.

Метамодель является экземпляром или конкретизацией мета-метамодели. Главная задача этого уровня — определить язык для спецификации моделей. Данный уровень является более конструктивным, чем предыдущий, поскольку обладает более развитой семантикой базовых понятий. Все основные понятия языка UML — это понятия уровня метамодели. Примеры таких понятий — класс, атрибут, операция, компонент, ассоциация и многие другие. Именно рассмотрению семантики и графической нотации понятий уровня метамодели посвящена данная книга.

Модель в контексте языка UML является экземпляром метамодели в том смысле, что любая конкретная модель системы должна использовать только понятия метамодели, конкретизировав их применительно к данной ситуации. Это уровень для описания информации о конкретной предметной области. Однако если для построения модели используются понятия языка UML, то необходима полная согласованность понятий уровня модели с базовыми понятиями языка UML уровня метамодели. Примерами понятий уровня модели могут служить, например, имена полей проектируемой базы данных, такие как имя и фамилия сотрудника, возраст, должность, адрес, телефон. При этом данные понятия используются лишь как имена соответствующих информационных атрибутов.

Конкретизация понятий модели происходит на уровне объектов. В настоящем контексте объект является экземпляром модели, поскольку содержит конкретную информацию относительно того, чему в действительности соответствуют те или иные понятия модели. Примером объекта может служить следующая запись в проектируемой базе данных: "Илья Петров, 30 лет, иллюзионист, ул. Невидимая, 10-20, 100-0000".

Описание семантики языка UML предполагает рассмотрение базовых понятий только уровня метамодели, который представляет собой лишь пример или частный случай уровня мета-метамодели. Метамодель UML является по своей сути, скорее логической моделью, чем физической или моделью реализации. Особенность логической модели заключается

в том, что она концентрирует внимание на декларативной или концептуальной семантике, опуская детали конкретной физической реализации моделей. При этом отдельные реализации, использующие данную логическую метамодель, должны быть согласованы с ее семантикой, а также поддерживать возможности импорта и экспорта отдельных логических моделей.

В то же время, логическая метамодель может быть реализована различными способами для обеспечения требуемого уровня производительности и надежности соответствующих инструментальных средств. В этом заключается недостаток логической модели, которая не содержит на уровне семантики требований, обязательных для ее эффективной последующей реализации. Однако согласованность метамодели с конкретными "моделями реализации является обязательной для всех разработчиков программных средств, обеспечивающих поддержку языка UML.

Метамодель языка UML имеет довольно сложную структуру, которая включает в себя порядка 90 метаклассов, более 100 метаассоциаций и почти 50 стереотипов, число которых возрастает с появлением новых версий языка. Чтобы справиться с этой сложностью языка UML, все его элементы организованы в логические пакеты. Поэтому рассмотрение языка UML на метамодельном уровне заключается в описании трех его наиболее общих логических блоков или пакетов: основные элементы, элементы поведения и общие механизмы.

В рамках языка UML все представления о модели сложной системы фиксируются в виде специальных графических конструкций, получивших название диаграмм. В терминах языка UML определены следующие виды диаграмм:

- Диаграмма вариантов использования (use case diagram)
- Диаграмма классов (class diagram)
- Диаграммы поведения (behavior diagrams)
  - Диаграмма состояний (statechart diagram)
  - Диаграмма деятельности (activity diagram)
  - Диаграммы взаимодействия (interaction diagrams)
- Диаграмма последовательности (sequence diagram)

- Диаграмма кооперации (collaboration diagram)
- Диаграммы реализации (implementation diagrams)
  - Диаграмма компонентов (component diagram)
  - Диаграмма развертывания (deployment diagram)

Из перечисленных выше диаграмм некоторые служат для обозначения двух и более других подвидов диаграмм. При этом в качестве самостоятельных представлений в языке UML используются следующие диаграммы:

1. Диаграмма вариантов использования
2. Диаграмма классов.
3. Диаграмма состояний.
4. Диаграмма деятельности.
5. Диаграмма последовательности.
6. Диаграмма кооперации.
7. Диаграмма компонентов.
8. Диаграмма развертывания.

Перечень этих диаграмм и их названия являются каноническими в том смысле, что представляют собой неотъемлемую часть графической нотации языка UML. Более того, процесс ООАП неразрывно связан с процессом построения этих диаграмм. При этом совокупность построенных таким образом диаграмм является самодостаточной в том смысле, что в них содержится вся информация, которая необходима для реализации проекта сложной системы.

Каждая из этих диаграмм детализирует и конкретизирует различные представления о модели сложной системы в терминах языка UML. При этом диаграмма вариантов использования представляет собой наиболее общую концептуальную модель сложной системы, которая является исходной для построения всех остальных диаграмм. Диаграмма классов является, по своей сути, логической моделью, отражающей статические аспекты структурного построения сложной системы.

Диаграммы поведения также являются разновидностями логической модели, которые отражают динамические аспекты функционирования сложной системы. И, наконец, диаграммы реализации служат для представления физических

компонентов сложной системы и поэтому относятся к ее физической модели. Таким образом, интегрированная модель сложной системы в нотации UML представляется в виде совокупности указанных выше диаграмм (рис. 2).



Рис. 2. Интегрированная модель сложной системы в нотации UML

Таким образом, процесс ООАП можно представить, как поуровневый спуск от наиболее общих моделей и представлений концептуального уровня к более частным и детальным представлениям логического и физического уровня. При этом на каждом из этапов ООАП данные модели последовательно дополняются все большим количеством деталей, что позволяет им более адекватно отражать различные аспекты конкретной реализации сложной системы

Концептуальная модель выражается в виде диаграмм прецедентов (use case diagram). Этот тип диаграмм служит для проведения итерационного цикла общей постановки задачи вместе с заказчиком.



Логическая модель позволяет определять два различных взгляда на системы: статический и динамический. Статический подход выражается диаграммами классов (class diagram).

Динамический подход описывается двумя типами диаграмм: диаграммами взаимодействия объектов; диаграммами последовательности взаимодействий.

Физическая модель задается компонентной диаграммой (component diagram), которая описывает распределение реализации классов по модулям, и диаграммой развертывания (deployment diagram).

## **3.2. Построение концептуальной модели.**

### **3.2.1. Разработка диаграмм вариантов использования**

Визуальное моделирование в UML можно представить, как некоторый процесс поуровневого спуска от наиболее общей и абстрактной концептуальной модели исходной системы к логической, а затем и к физической модели соответствующей программной системы. Для достижения этих целей вначале строится модель в форме, так называемой диаграммы вариантов использования (use case diagram), которая описывает функциональное назначение системы или, другими словами, то, что система будет делать в процессе своего функционирования. Диаграмма вариантов использования является исходным концептуальным представлением или концептуальной моделью системы в процессе ее проектирования и разработки.

Разработка диаграммы вариантов использования преследует цели:

- Определить общие границы и контекст моделируемой предметной области на начальных этапах проектирования системы.
- Сформулировать общие требования к функциональному поведению проектируемой системы.

- Разработать исходную концептуальную модель системы для ее последующей детализации в форме логических и физических моделей.
- Подготовить исходную документацию для взаимодействия разработчиков системы с ее заказчиками и пользователями.

Суть данной диаграммы состоит в следующем: проектируемая система представляется в виде множества сущностей или актеров, взаимодействующих с системой с помощью так называемых вариантов использования. При этом актером (actor) или действующим лицом называется любая сущность, взаимодействующая с системой извне. Это может быть человек, техническое устройство, программа или любая другая система, которая может служить источником воздействия на моделируемую систему так, как определит сам разработчик. В свою очередь, вариант использования (use case) служит для описания сервисов, которые система предоставляет актеру. Другими словами, каждый вариант использования определяет некоторый набор действий, совершаемый системой при диалоге с актером. При этом ничего не говорится о том, каким образом будет реализовано взаимодействие актеров с системой.

Между компонентами диаграммы вариантов использования могут существовать различные отношения, которые описывают взаимодействие экземпляров одних актеров и вариантов использования с экземплярами других актеров и вариантов. Один актер может взаимодействовать с несколькими вариантами использования. В этом случае этот актер обращается к нескольким сервисам данной системы. В свою очередь один вариант использования может взаимодействовать с несколькими актерами, предоставляя для всех них свой сервис. Следует заметить, что два варианта использования, определенные для одной и той же сущности, не могут взаимодействовать друг с другом, поскольку каждый из них самостоятельно описывает законченный вариант использования этой сущности. Более того, варианты использования всегда преду-

смаатривают некоторые сигналы или сообщения, когда взаимодействуют с актерами за пределами системы. В то же время могут быть определены другие способы для взаимодействия с элементами внутри системы.

В языке UML имеется несколько стандартных видов отношений между актерами и вариантами использования:

- Отношение ассоциации (association relationship)
- Отношение расширения (extend relationship)
- Отношение обобщения (generalization relationship)
- Отношение включения (include relationship)

При этом общие свойства вариантов использования могут быть представлены тремя различными способами, а именно с помощью отношений расширения, обобщения и включения.

### **Отношение ассоциации**

Отношение ассоциации является одним из фундаментальных понятий в языке UML и в той или иной степени используется при построении всех графических моделей систем в форме канонических диаграмм.

Применительно к диаграммам вариантов использования оно служит для обозначения специфической роли актера в отдельном варианте использования. Другими словами, ассоциация специфицирует семантические особенности взаимодействия актеров и вариантов использования в графической модели системы. Таким образом, это отношение устанавливает, какую конкретную роль играет актер при взаимодействии с экземпляром варианта использования.

### **Отношение расширения**

Отношение расширения определяет взаимосвязь экземпляров отдельного варианта использования с более общим вариантом, свойства которого определяются на основе способа совместного объединения данных экземпляров. В метамодели отношение расширения является направленным и указывает, что применительно к отдельным примерам некоторого варианта использования должны быть выполнены конкретные

условия, определенные для расширения данного варианта использования. Так, если имеет место отношение расширения от варианта использования А к варианту использования В, то это означает, что свойства экземпляра варианта использования В могут быть дополнены благодаря наличию свойств у расширенного варианта использования А.

Отношение расширения отмечает тот факт, что один из вариантов использования может присоединять к своему поведению некоторое дополнительное поведение, определенное для другого варианта использования. Данное отношение включает в себя некоторое условие и ссылки на точки расширения в базовом варианте использования. Чтобы расширение имело место, должно быть выполнено определенное условие данного отношения. Ссылки на точки расширения определяют те места в базовом варианте использования, в которые должно быть помещено соответствующее расширение при выполнении условия.

Один из вариантов использования может быть расширением для нескольких базовых вариантов, а также иметь в качестве собственных расширений несколько других вариантов. Базовый вариант использования может дополнительно никак не зависеть от своих расширений.

### **Отношение включения**

Отношение включения между двумя вариантами использования указывает, что некоторое заданное поведение для одного варианта использования включается в качестве составного компонента в последовательность поведения другого варианта использования. Данное отношение является направленным бинарным отношением в том смысле, что пара экземпляров вариантов использования всегда упорядочена в отношении включения.

Семантика этого отношения определяется следующим образом. Когда экземпляр первого варианта использования в процессе своего выполнения достигает точки включения в последовательность поведения экземпляра второго варианта использования, экземпляр первого варианта использования вы-

полняет последовательность действий, определяющую поведение экземпляра второго варианта использования, после чего продолжает выполнение действий своего поведения. При этом предполагается, что даже если экземпляр первого варианта использования может иметь несколько включаемых в себя экземпляров других вариантов, выполняемые ими действия должны закончиться к некоторому моменту, после чего должно быть продолжено выполнение прерванных действий экземпляра первого варианта использования в соответствии с заданным для него поведением.

Один вариант использования может быть включен в несколько других вариантов, а также включать в себя другие варианты. Включаемый вариант использования может быть независимым от базового варианта в том смысле, что он предоставляет последнему некоторое инкапсулированное поведение, детали реализации которого скрыты от последнего и могут быть легко перераспределены между несколькими включаемыми вариантами использования. Более того, базовый вариант может зависеть только от результатов выполнения включаемого в него поведения, но не от структуры включаемых в него вариантов.

Отношение включения, направленное от варианта использования А к варианту использования В, указывает, что каждый экземпляр варианта А включает в себя функциональные свойства, заданные для варианта В. Эти свойства специализируют поведение соответствующего варианта А на данной диаграмме.

### **3.2.2. Пример построения диаграммы вариантов использования**

В качестве примера рассмотрим процесс моделирования системы продажи товаров по каталогу, которая может быть использована при создании соответствующих информационных систем.

В качестве актеров данной системы могут выступать два субъекта, один из которых является продавцом, а другой — покупателем. Каждый из этих актеров взаимодействует с рассматриваемой системой продажи товаров по каталогу и является ее пользователем, т. е. они оба обращаются к соответствующему сервису "Оформить заказ на покупку товара".

Полученная диаграмма вариантов использования будет выглядеть следующим образом:



Рис. 3. Диаграмма вариантов использования

Приведенная выше диаграмма вариантов использования, в свою очередь, может быть детализирована далее с целью более глубокого уточнения предъявляемых к системе требований и конкретизации деталей ее последующей реализации. В рамках общей парадигмы ООАП подобная детализация может выполняться в двух основных направлениях.

С одной стороны, детализация может быть выполнена на основе установления дополнительных отношений типа отношения "обобщение-специализация" для уже имеющихся компонентов диаграммы вариантов использования. Так, в

рамках рассматриваемой системы продажи товаров может иметь самостоятельное значение и специфические особенности отдельная категория товаров — компьютеры. В этом случае диаграмма может быть дополнена вариантом использования "Оформить заказ на покупку компьютера" и актерами "Покупатель компьютера" и "Продавец компьютеров", которые связаны с соответствующими компонентами диаграммы отношением обобщения (рис. 4.).



Рис. 4. Уточненная диаграмма вариантов использования

Построение диаграммы вариантов использования является самым первым этапом процесса объектно-ориентированного анализа и проектирования, цель которого — представить совокупность требований к поведению проектируемой системы. Спецификация требований к проектируемой системе в форме диаграммы вариантов использования представляет собой самостоятельную модель, которая в языке UML получила название модели вариантов использования и

имеет свое специальное стандартное имя или стереотип "useCaseModel".

### **3.2.3. Рекомендации по разработке диаграмм вариантов использования**

Главное назначение диаграммы вариантов использования заключается в формализации функциональных требований к системе с помощью понятий соответствующего пакета и возможности согласования полученной модели с заказчиком на ранней стадии проектирования. Любой из вариантов использования может быть подвергнут дальнейшей декомпозиции на множество подвариантов использования отдельных элементов, которые образуют исходную сущность. Рекомендуются общее количество актеров в модели — не более 20, а вариантов использования — не более 50. В противном случае модель теряет свою наглядность и, возможно, заменяет собой одну из некоторых других диаграмм.

Семантика построения диаграммы вариантов использования должна определяться следующими особенностями рассмотренных выше элементов модели. Отдельный экземпляр варианта использования по своему содержанию является выполнением последовательности действий, которая инициализируется посредством экземпляра сообщения от экземпляра актера. В качестве отклика или ответной реакции на сообщение актера экземпляр варианта использования выполняет последовательность действий, установленную для данного варианта использования. Экземпляры актеров могут генерировать новые экземпляры сообщений для экземпляров вариантов использования.

Подобное взаимодействие будет продолжаться до тех пор, пока не закончится выполнение требуемой последовательности действий экземпляром варианта использования, и соответствующий экземпляр актера (и никакой другой) не получит требуемый экземпляр сервиса. Окончание взаимодействия означает отсутствие инициализации экземпляров сооб-



щений от экземпляров актеров для соответствующих экземпляров вариантов использования.

Варианты использования могут быть специфицированы в виде текста, а в последующем — с помощью операций и методов вместе с атрибутами, в виде графа деятельности, посредством автомата или любого другого механизма описания поведения, включающего предусловия и постусловия. Взаимодействие между вариантами использования и актерами может уточняться на диаграмме кооперации, когда описываются взаимосвязи между сущностью, содержащей эти варианты использования, и окружением или внешней средой этой сущности.

В случае, когда для представления иерархической структуры проектируемой системы используются подсистемы, система может быть определена в виде вариантов использования на всех уровнях. Отдельные подсистемы или классы могут выступать в роли таких вариантов использования. При этом вариант, соответствующий некоторому из этих элементов, в последующем может уточняться множеством более мелких вариантов использования, каждый из которых будет определять сервис элемента модели, содержащийся в сервисе исходной системы. Вариант использования в целом может рассматриваться как суперсервис для уточняющих его подвариантов, которые, в свою очередь, могут рассматриваться как подсервисы исходного варианта использования.

Функциональность, определенная для более общего варианта использования, полностью наследуется всеми вариантами нижних уровней. Однако следует заметить, что структура элемента-контейнера не может быть представлена вариантами использования, поскольку они могут представлять только функциональность отдельных элементов модели. Подчиненные варианты использования кооперируются для совместного выполнения суперсервиса варианта использования верхнего уровня. Эта кооперация также может быть представлена на диаграмме кооперации в виде совместных действий отдельных элементов модели.

Отдельные варианты использования нижнего уровня могут участвовать в нескольких кооперациях, т. е. играть определенную роль при выполнении сервисов нескольких вариантов верхнего уровня. Для отдельных таких коопераций могут быть определены соответствующие роли актеров, взаимодействующих с конкретными вариантами использования нижнего уровня. Эти роли будут играть актеры нижнего уровня модели системы. Хотя некоторые из таких актеров могут быть актерами верхнего уровня, это не противоречит принятым в языке UML семантическим правилам построения диаграмм вариантов использования. Более того, интерфейсы вариантов использования верхнего уровня могут полностью совпадать по своей структуре с соответствующими интерфейсами вариантов нижнего уровня.

Окружение вариантов использования нижнего уровня является самостоятельным элементом модели, который в свою очередь содержит другие элементы модели, определенные для этих вариантов использования. Таким образом, с точки зрения общего представления верхнего уровня взаимодействие между вариантами использования нижнего уровня определяет результат выполнения сервиса варианта верхнего уровня. Отсюда следует, что в языке UML вариант использования является элементом-контейнером.

Варианты использования классов соответствуют операциям этого класса, поскольку сервис класса является по существу выполнением операций данного класса. Некоторые варианты использования могут соответствовать применению только одной операции, в то время как другие — конечного множества операций, определенных в виде последовательности операций. В то же время одна операция может быть необходима для выполнения нескольких сервисов класса и поэтому будет появляться в нескольких вариантах использования этого класса.

Реализация варианта использования зависит от типа элемента модели, в котором он определен. Например, поскольку варианты использования класса определяются по-

средством операций этого класса, они реализуются соответствующими методами. С другой стороны, варианты использования подсистемы реализуются элементами, из которых состоит данная подсистема. Поскольку подсистема не имеет своего собственного поведения, все предлагаемые подсистемой сервисы должны представлять собой композицию сервисов, предлагаемых отдельными элементами этой подсистемы, т. е., в конечном итоге, классами. Эти элементы могут взаимодействовать друг с другом для совместного обеспечения требуемого поведения отдельного варианта использования. Такое совместное обеспечение требуемого поведения описывается специальным элементом языка UML — кооперация или сотрудничество. Здесь лишь отметим, что кооперации используются как для уточнения спецификаций в виде вариантов использования нижних уровней диаграммы, так и для описания особенностей их последующей реализации.

Если в качестве моделируемой сущности выступает система или подсистема самого верхнего уровня, то отдельные пользователи вариантов использования этой системы моделируются актерами. Такие актеры, являясь внутренними по отношению к моделируемым подсистемам нижних уровней, часто в явном виде не указываются, хотя и присутствуют неявно в модели подсистемы. Вместо этого варианты использования непосредственно обращаются к тем модельным элементам, которые содержат в себе подобные неявные актеры, т. е. экземпляры которых играют роли таких актеров при взаимодействии с вариантами использования. Эти модельные элементы могут содержаться в других пакетах или подсистемах. В последнем случае роли определяются в том пакете, к которому относится соответствующая подсистема.

С системно-аналитической точки зрения построение диаграммы вариантов использования специфицирует не только функциональные требования к проектируемой системе, но и выполняет исходную структуризацию предметной области. Последняя задача сочетает в себе не только следование техническим рекомендациям, но и является в некотором роде ис-

кусством, умением выделять главное в модели системы. Хотя рациональный унифицированный процесс не исключает итеративный возврат в последующем к диаграмме вариантов использования для ее модификации, не вызывает сомнений тот факт, что любая подобная модификация потребует, как по цепочке, изменений во всех других представлениях системы. Поэтому всегда необходимо стремиться к возможно более точному представлению модели именно в форме диаграммы вариантов использования.

Если же варианты использования применяются для спецификации части системы, то они будут эквивалентны соответствующим вариантам использования в модели подсистемы для части соответствующего пакета. Важно понимать, что все сервисы системы должны быть явно определены на диаграмме вариантов использования, и никаких других сервисов, которые отсутствуют на данной диаграмме, проектируемая система не может выполнять по определению. Более того, если для моделирования реализации системы используются сразу несколько моделей (например, модель анализа и модель проектирования), то множество вариантов использования всех пакетов системы должно быть эквивалентно множеству вариантов использования модели в целом.

### **3.3. Построение логической модели, отражающей статические аспекты работы сложной системы**

Диаграмма классов (class diagram) служит для представления статической структуры модели системы в терминологии классов объектно-ориентированного программирования. Диаграмма классов может отражать, в частности, различные взаимосвязи между отдельными сущностями предметной области, такими как объекты и подсистемы, а также описывает их внутреннюю структуру и типы отношений. На данной диаграмме не указывается информация о временных аспектах функционирования системы. С этой точки зрения диаграмма

классов является дальнейшим развитием концептуальной модели проектируемой системы.

Диаграмма классов представляет собой некоторый граф, вершинами которого являются элементы типа "классификатор", которые связаны различными типами структурных отношений. Следует заметить, что диаграмма классов может также содержать интерфейсы, пакеты, отношения и даже отдельные экземпляры, такие как объекты и связи. Когда говорят о данной диаграмме, имеют в виду статическую структурную модель проектируемой системы. Поэтому диаграмму классов принято считать графическим представлением таких структурных взаимосвязей логической модели системы, которые не зависят или инвариантны от времени.

Диаграмма классов состоит из множества элементов, которые в совокупности отражают декларативные знания о предметной области. Эти знания интерпретируются в базовых понятиях языка UML, таких как классы, интерфейсы и отношения между ними и их составляющими компонентами. При этом отдельные компоненты этой диаграммы могут образовывать пакеты для представления более общей модели системы. Если диаграмма классов является частью некоторого пакета, то ее компоненты должны соответствовать элементам этого пакета, включая возможные ссылки на элементы из других пакетов.

В общем случае пакет статической структурной модели может быть представлен в виде одной или нескольких диаграмм классов. Декомпозиция некоторого представления на отдельные диаграммы выполняется с целью удобства и графической визуализации структурных взаимосвязей предметной области. При этом компоненты диаграммы соответствуют элементам статической семантической модели. Модель системы, в свою очередь, должна быть согласована с внутренней структурой классов, которая описывается на языке UML.

Класс (class) в языке UML служит для обозначения множества объектов, которые обладают одинаковой структурой, поведением и отношениями с объектами из других клас-

сов. Графически класс изображается в виде прямоугольника, который дополнительно может быть разделен горизонтальными линиями на разделы или секции (рис. 30). В этих разделах могут указываться имя класса, атрибуты (переменные) и операции (методы).

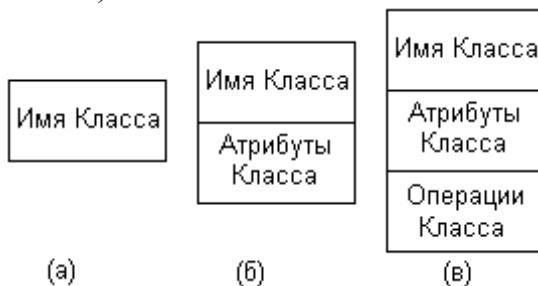


Рис. 5. Графическое изображение класса на диаграмме классов

Обязательным элементов обозначения класса является его имя. На начальных этапах разработки диаграммы отдельные классы могут обозначаться простым прямоугольником с указанием только имени соответствующего класса (рис. 5, а). По мере проработки отдельных компонентов диаграммы описания классов дополняются атрибутами (рис. 5, б) и операциями (рис. 5, в).

Предполагается, что окончательный вариант диаграммы содержит наиболее полное описание классов, которые состоят из трех разделов или секций. Иногда в обозначениях классов используется дополнительный четвертый раздел, в котором приводится семантическая информация справочного характера или явно указываются исключительные ситуации.

Даже если секция атрибутов и операций является пустой, в обозначении класса она выделяется горизонтальной линией, чтобы сразу отличить класс от других элементов языка UML. Примеры графического изображения классов на диаграмме классов приведены на рис. 6. В первом случае для класса "Прямоугольник" (рис. 6, а) указаны только его атрибуты — точки на координатной плоскости, которые опреде-

ляют его расположение. Для класса "Окно" (рис. 6, б) указаны только его операции, секция атрибутов оставлена пустой. Для класса "Счет" (рис. 6, в) дополнительно изображена четвертая секция, в которой указано исключение — отказ от обработки просроченной кредитной карточки.

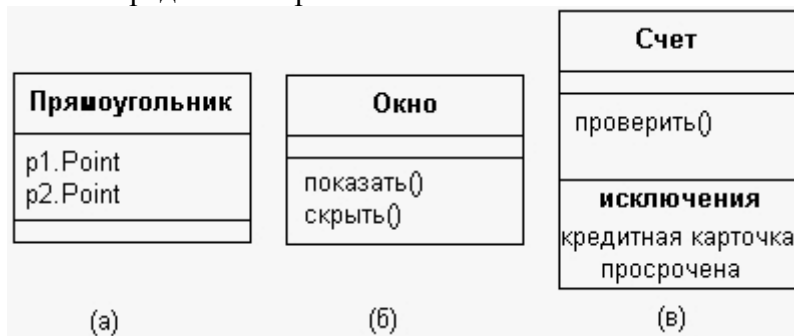


Рис. 6. Примеры графического изображения классов на диаграмме

Имя класса должно быть уникальным в пределах пакета, который описывается некоторой совокупностью диаграмм классов (возможно, одной диаграммой). Оно указывается в первой верхней секции прямоугольника. В дополнение к общему правилу наименования элементов языка UML, имя класса записывается по центру секции имени полужирным шрифтом и должно начинаться с заглавной буквы. Рекомендуется в качестве имен классов использовать существительные, записанные по практическим соображениям без пробелов. Необходимо помнить, что именно имена классов образуют словарь предметной области при ООАП.

В первой секции обозначения класса могут находиться ссылки на стандартные шаблоны или абстрактные классы, от которых образован данный класс и, соответственно, от которых он наследует свойства и методы. В этой секции может приводиться информация о разработчике данного класса и статус состояния разработки, а также могут записываться и другие общие свойства этого класса, имеющие отношение к

другим классам диаграммы или стандартным элементам языка UML.

Примерами имен классов могут быть такие существительные, как "Сотрудник", "Компания", "Руководитель", "Клиент", "Продавец", "Менеджер", "Офис", "Студент" и многие другие, имеющие непосредственное отношение к моделируемой предметной области и функциональному назначению проектируемой системы.

Класс может не иметь экземпляров или объектов. В этом случае он называется абстрактным классом, а для обозначения его имени используется наклонный шрифт (курсив). В языке UML принято общее соглашение о том, что любой текст, относящийся к абстрактному элементу, записывается курсивом. Данное обстоятельство является семантическим аспектом описания соответствующих элементов языка UML.

В некоторых случаях необходимо явно указать, к какому пакету относится тот или иной класс. Для этой цели используется специальный символ разделитель — двойное двоеточие "::". Синтаксис строки имени класса в этом случае будет следующий <Имя\_пакета>::<Имя\_класса>. Другими словами, перед именем класса должно быть явно указано имя пакета, к которому его следует отнести. Например, если определен пакет с именем "Банк", то класс "Счет" в этом банке может быть записан в виде: "Банк::Счет".

## **Атрибуты класса**

Во второй сверху секции прямоугольника класса записываются его атрибуты (attributes) или свойства. В языке UML принята определенная стандартизация записи атрибутов класса, которая подчиняется некоторым синтаксическим правилам. Каждому атрибуту класса соответствует отдельная строка текста, которая состоит из квантора видимости атрибута, имени атрибута, его кратности, типа значений атрибута и, возможно, его исходного значения.



## Методы класса

В третьей сверху секции прямоугольника записываются операции или методы класса. Операция (operation) представляет собой некоторый сервис, предоставляющий каждый экземпляр класса по определенному требованию. Совокупность операций характеризует функциональный аспект поведения класса. Запись операций класса в языке UML также стандартизована и подчиняется определенным синтаксическим правилам. При этом каждой операции класса соответствует отдельная строка, которая состоит из квантора видимости операции, имени операции, выражения типа, возвращаемого операцией значения

## Отношения между классами

Кроме внутреннего устройства или структуры классов на соответствующей диаграмме указываются различные отношения между классами. При этом совокупность типов таких отношений фиксирована в языке UML и предопределена семантикой этих типов отношений. Базовыми отношениями или связями в языке UML являются:

- Отношение зависимости (dependency relationship)
- Отношение ассоциации (association relationship)
- Отношение обобщения (generalization relationship)
- Отношение реализации (realization relationship)

Каждое из этих отношений имеет собственное графическое представление на диаграмме, которое отражает взаимосвязи между объектами соответствующих классов.

Рассмотрим сходства и различия между следующими классами: цветы, маргаритки, красные розы, желтые розы, лепестки и божьи коровки. Мы можем заметить следующее:

- Маргаритка - цветок.
- Роза - (другой) цветок.
- Красная и желтая розы - розы.
- Лепесток является частью обоих видов цветов.

- Божьи коровки питаются вредителями, поражающими некоторые цветы.

Из этого простого примера следует, что классы, как и объекты, не существуют изолированно. В каждой проблемной области ключевые абстракции взаимодействуют многими интересными способами, что мы и должны отразить в проекте.

Отношения между классами могут означать одно из двух. Во-первых, у них может быть что-то общее. Например, и маргаритки, и розы - это разновидности цветов: и те, и другие имеют ярко окрашенные лепестки, испускают аромат и так далее. Во-вторых, может быть какая-то семантическая связь. Например, красные розы больше похожи на желтые розы, чем на маргаритки. Но между розами и маргаритками больше общего, чем между цветами и лепестками. Также существует симбиотическая связь между цветами и божьими коровками: божьи коровки защищают цветы от вредителей, которые, в свою очередь, служат пищей божьим коровкам. UML диаграмма отношений между этими классами представлена на рис. 7.

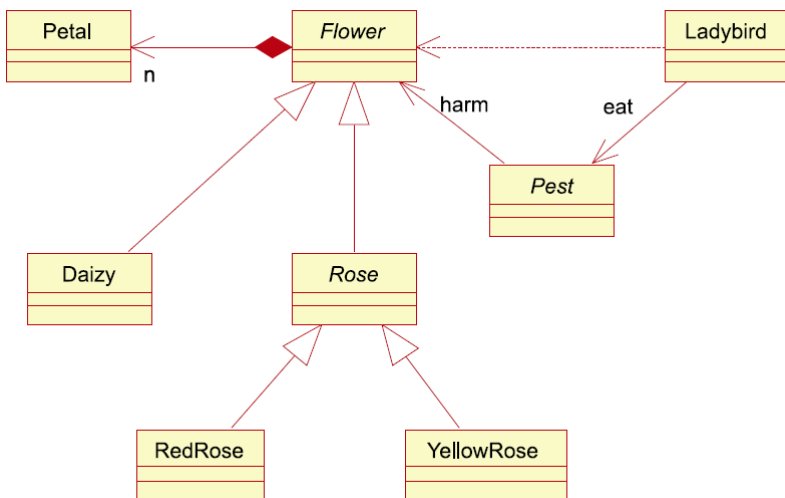


Рис. 7. Отношения между классами

Рассмотрим основные типы отношений между классами на данном примере. Во-первых, это отношение "обобщение/специализация" (общее и частное), известное как "is-a". Розы суть цветы, что значит: розы являются специализированным частным случаем, подклассом более общего класса "цветы". Во вторых, это отношение "целое/ часть", известное как "part of". Так, лепестки являются частью цветов. В третьих, это семантические, смысловые отношения, ассоциации. Например, божьи коровки ассоциируются с цветами - хотя, казалось бы, что у них общего. Или вот: розы и свечи - и то, и другое можно использовать для украшения стола.

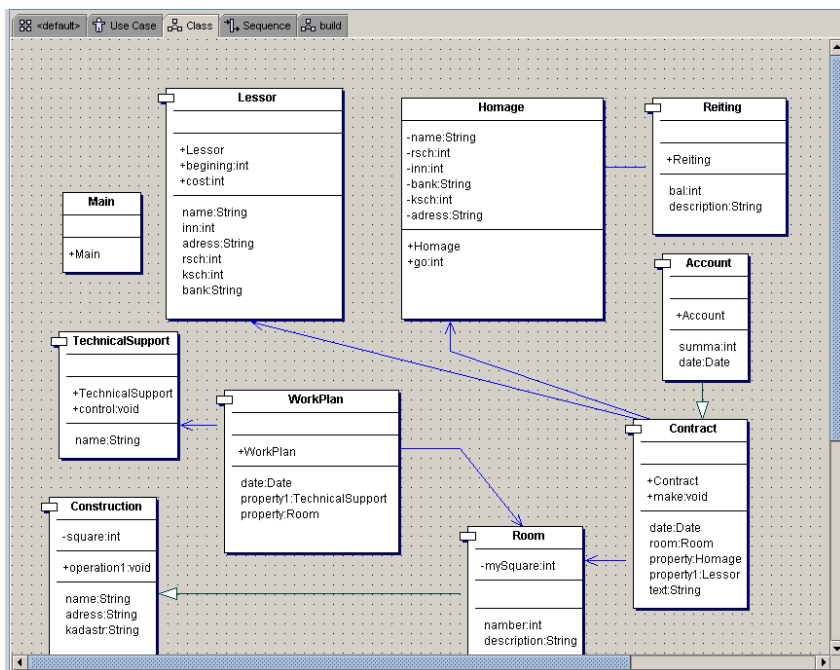


Рис. 8. Пример графического изображения диаграммы классов с различными типами отношений между классами, разработанной в среде Together

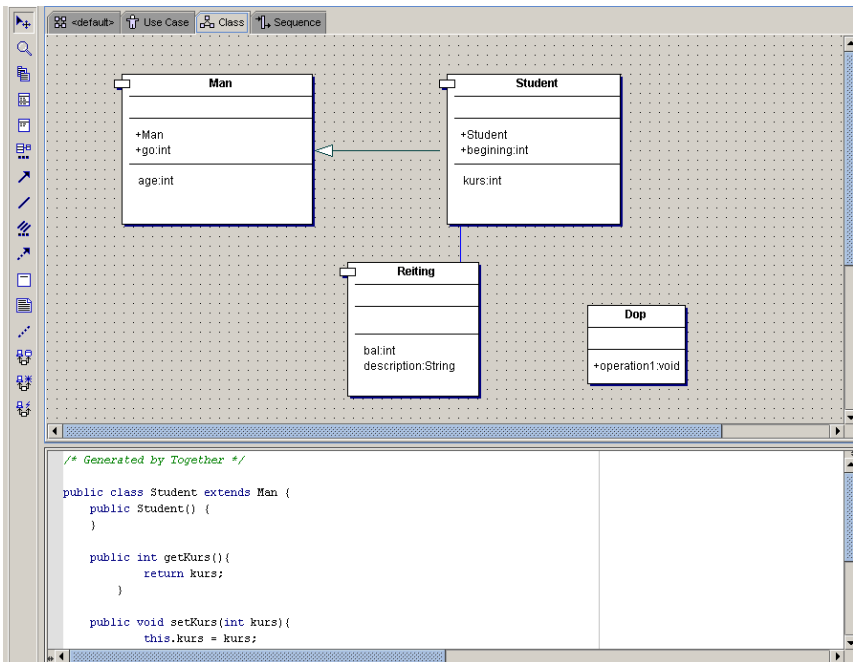


Рис. 9. Пример графического изображения диаграммы классов с элементами сгенерированного кода, разработанной в среде Together

Шаблоны классов, автоматически сгенерированные средой Together представлены ниже:

```
/* Generated by Together */
```

```

public class Reiting {
    public int getBal(){
        return bal;
    }

    public void setBal(int bal){
        this.bal = bal;
    }
}

```

```
public String getDescription(){
    return description;
}

public void setDescription(String description){
    this.description = description;
}

private int bal;
private String description;
}
```

```
/* Generated by Together */
```

```
public class Student extends Man {
    public Student() {
    }

    public int getKurs(){
        return kurs;
    }

    public void setKurs(int kurs){
        this.kurs = kurs;
    }

    private int kurs;
    private Reiting lnkClass1;
}
```

```
/* Generated by Together */
```

```
public class Man {
    private int age;
```

```

public Man(int age) {
    this.age=age;
}

public int getAge()
{
    return age;
}
public void setAge(int age)
{
}
public int go()
{
}
}

```

Процесс разработки диаграммы классов занимает центральное место в ООАП сложных систем. От умения правильно выбрать классы и установить между ними взаимосвязи часто зависит не только успех процесса проектирования, но и производительность выполнения программы. Как показывает практика ООП, каждый программист в своей работе стремится в той или иной степени использовать уже накопленный личный опыт при разработке новых проектов. Это обусловлено желанием свести новую задачу к уже решенным, чтобы иметь возможность использовать не только проверенные фрагменты программного кода, но и отдельные компоненты в целом (библиотеки компонентов).

Такой стереотипный подход позволяет существенно сократить сроки реализации проекта, однако приемлем лишь в том случае, когда новый проект концептуально и технологически не слишком отличается от предыдущих. В противном случае платой за сокращение сроков проекта может стать его реализация на устаревшей технологической базе. Что касается собственно объектной структуризации предметной области,

то здесь уместно придерживаться тех рекомендаций, которые накоплены в ООП.

При определении классов, атрибутов и операций и задании их имен и типов перед отечественными разработчиками всегда встает невольный вопрос: какой из языков использовать в качестве естественного, русский или английский? С одной стороны, использование родного языка для описания модели является наиболее естественным способом ее представления и в наибольшей степени отражает коммуникативную функцию модели системы. С другой стороны, разработка модели является лишь одним из этапов разработки соответствующей системы, а применение инструментальных средств для ее реализации в абсолютном большинстве случаев требует использования англоязычных терминов.

Отвечая на поставленный выше вопрос, следует отметить, что наиболее целесообразно придерживаться следующих рекомендаций. При построении диаграммы вариантов использования, являющейся наиболее общей концептуальной моделью проектируемой системы, применение русскоязычных терминов является не только оправданным с точки зрения описания структуры предметной области, но и эффективным с точки зрения коммуникативного взаимодействия с заказчиком и пользователями. При построении остальных типов диаграмм следует придерживаться разумного компромисса.

В частности, на начальных этапах разработки диаграмм целесообразность использования русскоязычных терминов вполне очевидна и оправдана. Однако, по мере готовности графической модели для реализации в виде программной системы и передачи ее для дальнейшей работы программистам, акцент может смещаться в сторону использования англоязычных терминов, которые в той или иной степени отражают особенности языка программирования, на котором предполагается реализация данной модели.

Более того, использование CASE-инструментариев для автоматизации ООАП, чаще всего, накладывает свои собственные требования на язык спецификации моделей. Именно

по этой причине большинство примеров в литературе даются в англоязычном представлении, а при их переводе на русский может быть утрачена не только точность формулировок, но и семантика соответствующих понятий.

После разработки диаграммы классов процесс ООАП может быть продолжен в двух направлениях. С одной стороны, если поведение системы тривиально, то можно приступить к разработке диаграмм кооперации и компонентов. Однако для сложных динамических систем поведение представляет важнейший аспект их функционирования. Детализация поведения осуществляется последовательно при разработке диаграмм состояний, последовательности и деятельности.

### **3.4. Построение логической модели, отражающей динамические аспекты работы сложной системы**

#### **3.4.1. Разработка диаграмм последовательности (sequence diagram)**

Для моделирования взаимодействия объектов в языке UML используются соответствующие диаграммы взаимодействия. Говоря об этих диаграммах, имеют в виду два аспекта взаимодействия. Во-первых, взаимодействия объектов можно рассматривать во времени, и тогда для представления временных особенностей передачи и приема сообщений между объектами используется диаграмма последовательности. Этот вид канонических диаграмм является предметом изучения настоящей главы.

Ранее, при изучении диаграмм состояния и деятельности, было отмечено одно немаловажное обстоятельство. Хотя рассмотренные диаграммы и используются для спецификации динамики поведения систем, время в явном виде в них не присутствует. Однако временной аспект поведения может иметь существенное значение при моделировании синхронных процессов, описывающих взаимодействия объектов. Именно для



этой цели в языке UML используются диаграммы последовательности.

Во-вторых, можно рассматривать структурные особенности взаимодействия объектов. Для представления структурных особенностей передачи и приема сообщений между объектами используется диаграмма кооперации.

## **Объекты**

На диаграмме последовательности изображаются исключительно те объекты, которые непосредственно участвуют во взаимодействии и не показываются возможные статические ассоциации с другими объектами. Для диаграммы последовательности ключевым моментом является именно динамика взаимодействия объектов во времени. При этом диаграмма последовательности имеет как бы два измерения. Одно — слева направо в виде вертикальных линий, каждая из которых изображает линию жизни отдельного объекта, участвующего во взаимодействии. Графически каждый объект изображается прямоугольником и располагается в верхней части своей линии жизни (рис. 10). Внутри прямоугольника записываются имя объекта и имя класса, разделенные двоеточием. При этом вся запись подчеркивается, что является признаком объекта, который, как известно, представляет собой экземпляр класса.

Не исключается ситуация, когда имя объекта может отсутствовать на диаграмме последовательности. В этом случае указывается только имя класса, а сам объект считается анонимным.

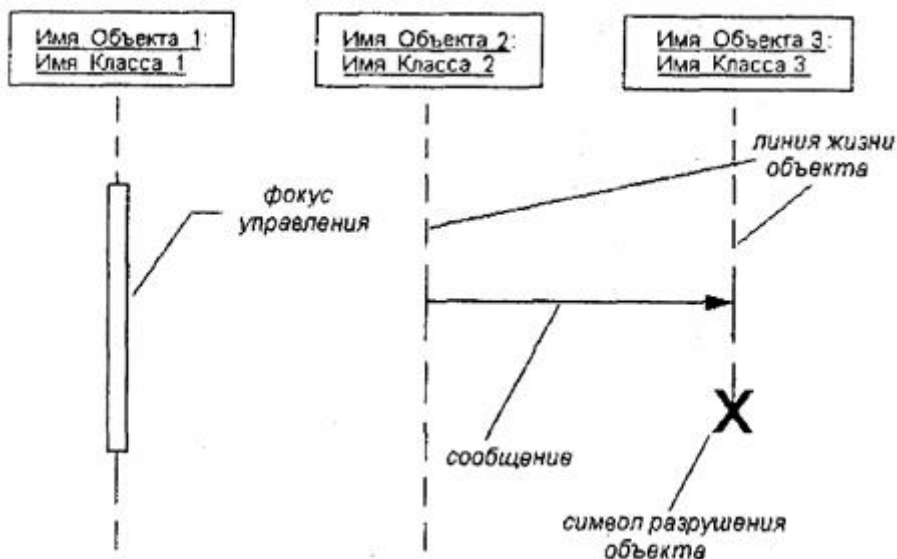


Рис. 10. Различные графические примитивы диаграммы последовательности

Крайним слева на диаграмме изображается объект, который является инициатором взаимодействия (объект 1 на рис. 10). Правее изображается другой объект, который непосредственно взаимодействует с первым. Таким образом, все объекты на диаграмме последовательности образуют некоторый порядок, определяемый степенью активности этих объектов при взаимодействии друг с другом.

Второе измерение диаграммы последовательности — вертикальная временная ось, направленная сверху вниз. Начальному моменту времени соответствует самая верхняя часть диаграммы. При этом взаимодействия объектов реализуются посредством сообщений, которые посылаются одними объектами другим. Сообщения изображаются в виде горизонтальных стрелок с именем сообщения и также образуют порядок по времени своего возникновения. Другими словами, сообщения, расположенные на диаграмме последовательности выше, инициируются раньше тех, которые расположены ни-

же. При этом масштаб на оси времени не указывается, поскольку диаграмма последовательности моделирует лишь временную упорядоченность взаимодействий типа "раньше-позже".

### **Линия жизни объекта**

Линия жизни объекта (object lifeline) изображается пунктирной вертикальной линией, ассоциированной с единственным объектом на диаграмме последовательности. Линия жизни служит для обозначения периода времени, в течение которого объект существует в системе и, следовательно, может потенциально участвовать во всех ее взаимодействиях. Если объект существует в системе постоянно, то и его линия жизни должна продолжаться по всей плоскости диаграммы последовательности от самой верхней ее части до самой нижней (объекты 1 и 2 на рис. 10).

Отдельные объекты, выполнив свою роль в системе, могут быть уничтожены (разрушены), чтобы освободить занимаемые ими ресурсы. Для таких объектов линия жизни обрывается в момент его уничтожения. Для обозначения момента уничтожения объекта в языке UML используется специальный символ в форме латинской буквы "X" (объект 3 на рис. 35). Ниже этого символа пунктирная линия не изображается, поскольку соответствующего объекта в системе уже нет, и этот объект должен быть исключен из всех последующих взаимодействий.

Вовсе не обязательно создавать все объекты в начальный момент времени. Отдельные объекты в системе могут создаваться по мере необходимости, существенно экономя ресурсы системы и повышая ее производительность. В этом случае прямоугольник такого объекта изображается не в верхней части диаграммы последовательности, а в той ее части, которая соответствует моменту создания объекта (объект 6 на рис. 11). При этом прямоугольник объекта вертикально располагается в том месте диаграммы, которое по оси времени

совпадает с моментом его возникновения в системе. Очевидно, объект обязательно создается со своей линией жизни и, возможно, с фокусом управления.

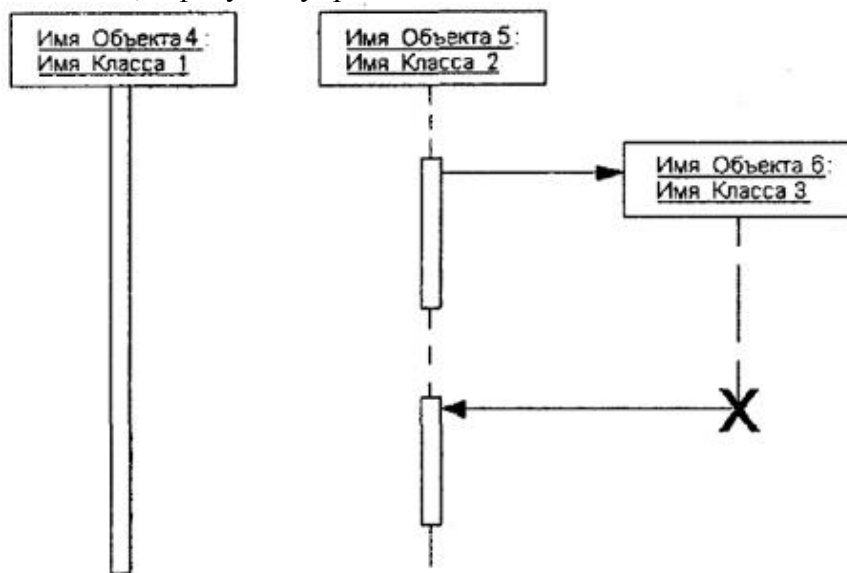


Рис. 11. Графическое изображение различных вариантов линий жизни и фокусов управления объектов

### Фокус управления

В процессе функционирования объектно-ориентированных систем одни объекты могут находиться в активном состоянии, непосредственно выполняя определенные действия или в состоянии пассивного ожидания сообщений от других объектов. Чтобы явно выделить подобную активность объектов, в языке UML применяется специальное понятие, получившее название фокуса управления (focus of control). Фокус управления изображается в форме вытянутого узкого прямоугольника (см. объект 1 на рис. 35), верхняя сторона которого обозначает начало получения фокуса управления объектом (начало активности), а ее нижняя сторона — окончание фокуса управления (окончание активности). Этот

прямоугольник располагается ниже обозначения соответствующего объекта и может заменять его линию жизни (объект 4 на рис. 11), если на всем ее протяжении он является активным.

С другой стороны, периоды активности объекта могут чередоваться с периодами его пассивности или ожидания. В этом случае у такого объекта имеются несколько фокусов управления (объект 5 на рис. 11). Важно понимать, что получить фокус управления может только существующий объект, у которого в этот момент имеется линия жизни. Если же некоторый объект был уничтожен, то вновь возникнуть в системе он уже не может. Вместо него лишь может быть создан другой экземпляр этого же класса, который, строго говоря, будет являться другим объектом.

В отдельных случаях инициатором взаимодействия в системе может быть актер или внешний пользователь. В этом случае актер изображается на диаграмме последовательности самым первым объектом слева со своим фокусом управления (рис. 12). Чаще всего актер и его фокус управления будут существовать в системе постоянно, отмечая характерную для пользователя активность в инициировании взаимодействий с системой. При этом сам актер может иметь собственное имя либо оставаться анонимным.

Иногда некоторый объект может инициировать рекурсивное взаимодействие с самим собой. Речь идет о том, что наличие во многих языках программирования специальных средств построения рекурсивных процедур требует визуализации соответствующих понятий в форме графических примитивов. На диаграмме последовательности рекурсия обозначается небольшим прямоугольником, присоединенным к правой стороне фокуса управления того объекта, для которого изображается это рекурсивное взаимодействие (объект 7 на рис. 12).

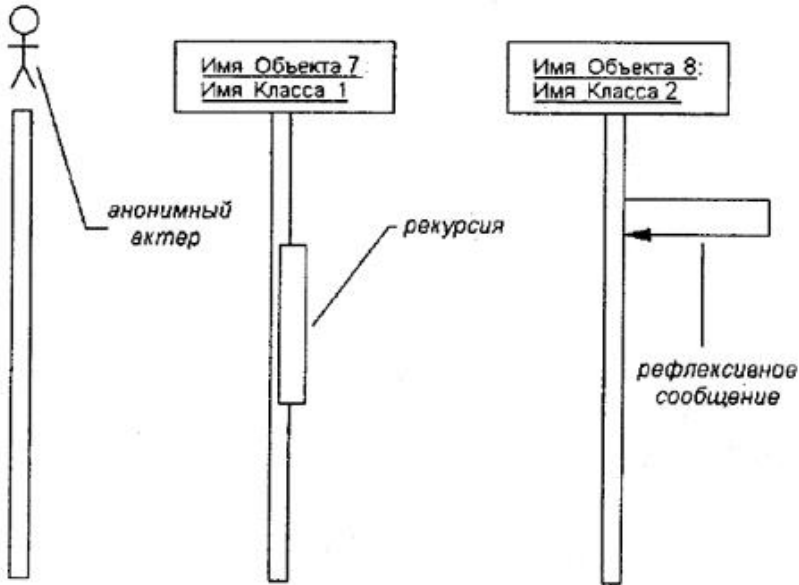


Рис. 12. Графическое изображение актера, рекурсии и рефлексивного сообщения на диаграмме последовательности

## Сообщения

Как было отмечено выше, цель взаимодействия в контексте языка UML заключается в том, чтобы специфицировать коммуникацию между множеством взаимодействующих объектов. Каждое взаимодействие описывается совокупностью сообщений, которыми участвующие в нем объекты обмениваются между собой. В этом смысле сообщение (message) представляет собой законченный фрагмент информации, который отправляется одним объектом другому. При этом прием сообщения инициирует выполнение определенных действий, направленных на решение отдельной задачи тем объектом, которому это сообщение отправлено.

Таким образом, сообщения не только передают некоторую информацию, но и требуют или предполагают от принимающего объекта выполнения ожидаемых действий. Сообщения могут инициировать выполнение операций объектом со-

ответствующего класса, а параметры этих операций передаются вместе с сообщением. На диаграмме последовательности все сообщения упорядочены по времени своего возникновения в моделируемой системе.

В таком контексте каждое сообщение имеет направление от объекта, который инициирует и отправляет сообщение, к объекту, который его получает. Иногда отправителя сообщения называют клиентом, а получателя — сервером. При этом сообщение от клиента имеет форму запроса некоторого сервиса, а реакция сервера на запрос после получения сообщения может быть связана с выполнением определенных действий или передачи клиенту необходимой информации тоже в форме сообщения.

В языке UML могут встречаться несколько разновидностей сообщений, каждое из которых имеет свое графическое изображение.

- Первая разновидность сообщения является наиболее распространенной и используется для вызова процедур, выполнения операций или обозначения отдельных вложенных потоков управления. Начало этой стрелки всегда соприкасается с фокусом управления или линией жизни того объекта-клиента, который инициирует это сообщение. Конец стрелки соприкасается с линией жизни того объекта, который принимает это сообщение и выполняет в ответ определенные действия. При этом принимающий объект зачастую получает и фокус управления, становясь активным.

- Вторая разновидность сообщения используется для обозначения простого (не вложенного) потока управления. Каждая такая стрелка указывает на прогресс одного шага потока. При этом соответствующие сообщения обычно являются асинхронными, т. е. могут возникать в произвольные моменты времени. Передача такого сообщения обычно сопровождается получением фокуса управления объектом, его принявшим.

- Третья разновидность явно обозначает асинхронное сообщение между двумя объектами в некоторой про-

цедурной последовательности. Примером такого сообщения может служить прерывание операции при возникновении исключительной ситуации. В этом случае информация о такой ситуации передается вызываемому объекту для продолжения процесса дальнейшего взаимодействия.

- Наконец, последняя разновидность сообщения используется для возврата из вызова процедуры. Примером может служить простое сообщение о завершении некоторых вычислений без предоставления результата расчетов объекту-клиенту. В процедурных потоках управления эта стрелка может быть опущена, поскольку ее наличие неявно предполагается в конце активизации объекта. В то же время считается, что каждый вызов процедуры имеет свою пару — возврат вызова. Для непроцедурных потоков управления, включая параллельные и асинхронные сообщения, стрелка возврата должна указываться явным образом.

Обычно сообщения изображаются горизонтальными стрелками, соединяющими линии жизни или фокусы управления двух объектов на диаграмме последовательности. При этом неявно предполагается, что время передачи сообщения достаточно мало по сравнению с процессами выполнения действий объектами. Считается также, что за время передачи сообщения с соответствующими объектами не может произойти никаких событий. Другими словами, состояния объектов остаются без изменения. Если же это предположение не может быть признано справедливым, то стрелка сообщения изображается под некоторым наклоном, так чтобы конец стрелки располагался ниже ее начала.

В отдельных случаях объект может посылать сообщения самому себе, иницируя так называемые рефлексивные сообщения (объект 8 на рис. 37). Такие сообщения изображаются прямоугольником со стрелкой, начало и конец которой совпадают. Подобные ситуации возникают, например, при обработке нажатий на клавиши клавиатуры при вводе текста в редактируемый документ, при наборе цифр номера телефона абонента.



Таким образом, в языке UML каждое сообщение ассоциируется с некоторым действием, которое должно быть выполнено принявшим его объектом. При этом действие может иметь некоторые аргументы или параметры, в зависимости от конкретных значений которых может быть получен различный результат. Соответствующие параметры будет иметь и вызывающее это действие сообщение. Более того, значения параметров отдельных сообщений могут содержать условные выражения, образуя ветвление или альтернативные пути основного потока управления.

Построение диаграммы последовательности сводится к добавлению или удалению отдельных объектов и сообщений, а также к их спецификации. Доступ к спецификации этих элементов организован либо через контекстное меню, либо через пункт меню Browse - Specification (Браузер - Спецификация). При добавлении сообщений на диаграмму последовательности они получают по умолчанию свой номер в последовательности.

### **Пример построения диаграммы последовательности**

В качестве примера рассмотрим построение диаграммы последовательности для моделирования процесса телефонного разговора с использованием обычной телефонной сети. Объектами в этом примере являются: два абонента, а и Б, два телефонных аппарата end, коммутатор и сам разговор как объект моделирования. При этом как коммутатор, так и разговор являются анонимными объектами.

На первом этапе располагаем выбранные объекты на предполагаемой диаграмме (рис. 13). Заметим, что абонентов мы будем рассматривать как актеров, причем первый из них — а — играет активную роль, а второй — б — пассивную роль. Поэтому первый получает фокус управления сразу после своего появления в системе, а второй имеет только линию жизни. Коммутатор также имеет постоянную активность, что изображается его фокусом управления. Разговор как объект

появляется только после установки соединения и уничтожается с его прекращением. Поэтому он будет изображен позже на этой же диаграмме последовательности.

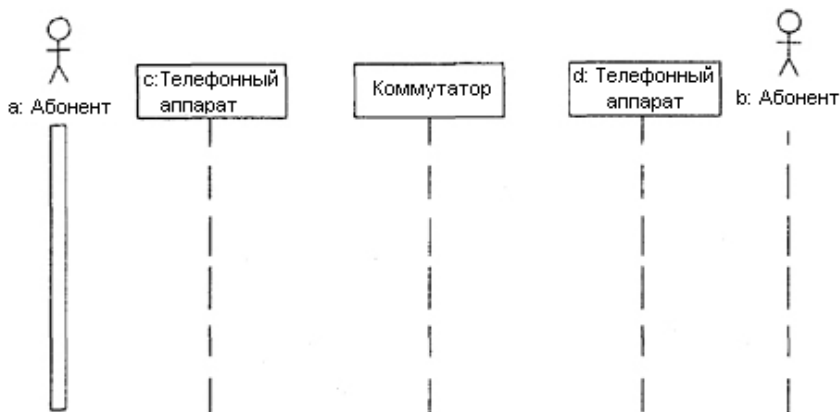


Рис. 13. Начальный фрагмент диаграммы последовательности для моделирования телефонного разговора

Процесс взаимодействия в этой системе начинается с поднятия трубки телефонного аппарата первым абонентом. Тем самым он посылает сообщение телефонному аппарату с, которое переводит этот аппарат в активное состояние и вызывает действие — подачу тонового сигнала в телефонную трубку для первого абонента. Следующее действие также инициируется первым абонентом — набор цифр телефонного номера. Это представлено в форме итеративного сообщения со знаком "\*" слева от его имени.

Заметим, что поднятие телефонной трубки и набор цифр номера являются физическими действиями и поэтому изображаются в форме простых асинхронных сообщений. После набора цифр'номера телефона аппарат с рекурсивно вызывает процедуру отправки коммутационных импульсов на коммутатор. Последний инициирует создание нового объекта в моделируемой системе — телефонного разговора. Дополненный фрагмент диаграммы последовательности изображен на рис. 14.

После создания анонимный объект "разговор" сразу получает фокус активности и посылает сообщение телефонному аппарату d на выполнение действия — звонка вызова. При этом второй абонент снимает трубку (асинхронное сообщение), тем самым устанавливается прямое соединение между абонентами a и b. После того как абоненты опустят трубки, разговор заканчивается. Тем самым объект "разговор" уничтожается. Окончательный вариант диаграммы последовательности может содержать некоторые временные ограничения и комментарии (рис. 15). Назначение отдельных сообщений соответствует рассмотренным действиям.

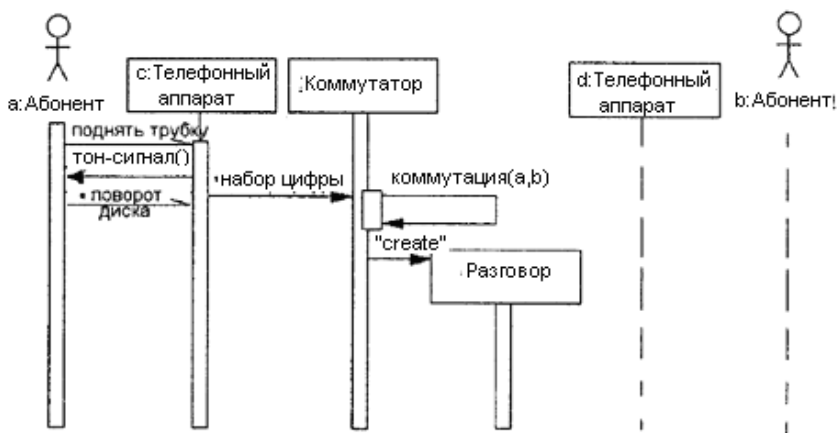


Рис. 14. Дополненный фрагмент диаграммы последовательности для моделирования телефонного разговора

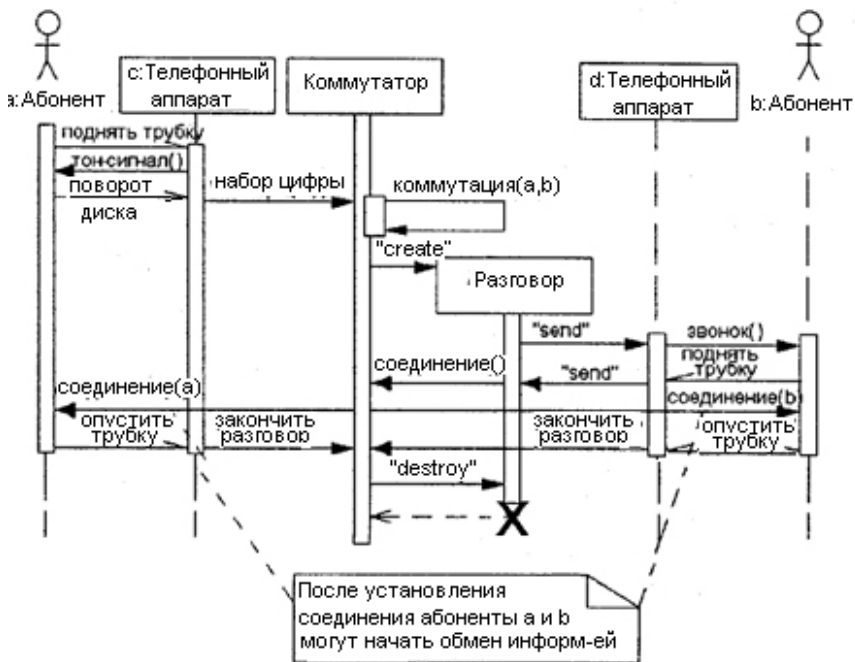


Рис. 15. Окончательный вариант диаграммы последовательности для моделирования телефонного разговора

Ниже приводится пример построенной диаграммы последовательности в среде Together (рис. 16).

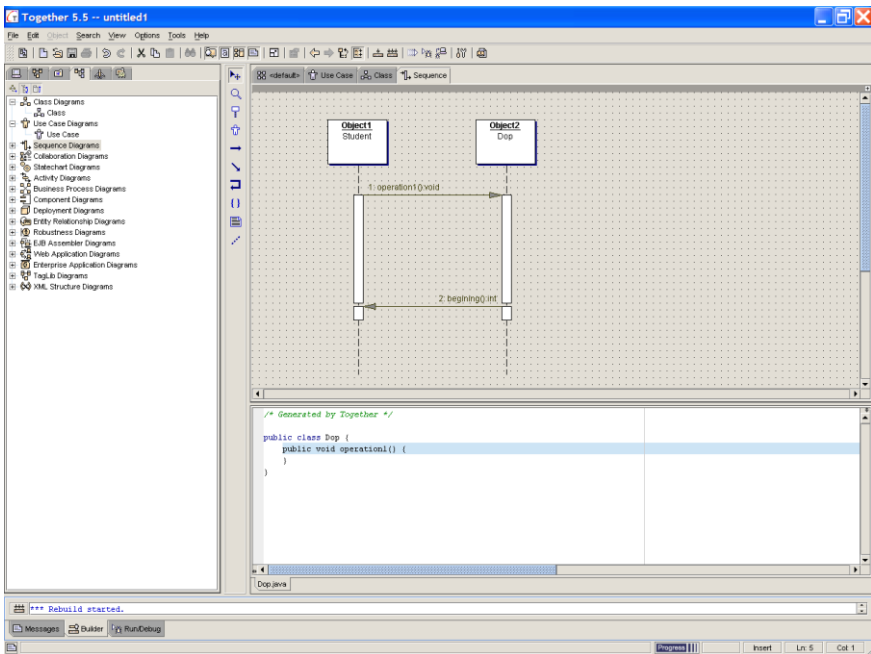


Рис. 16. Пример диаграммы последовательности

Как уже отмечалось, построение диаграммы последовательности целесообразно начинать с выделения из всей совокупности тех и только тех классов, объекты которых участвуют в моделируемом взаимодействии. После этого все объекты наносятся на диаграмму с соблюдением некоторого порядка инициализации сообщений. Здесь необходимо установить, какие объекты будут существовать постоянно, а какие временно — только на период выполнения ими требуемых действий.

Когда объекты визуализированы, можно приступить к спецификации сообщений. При этом следует учитывать те роли, которые играют сообщения в системе. При необходимости уточнения этих ролей надо использовать их разновидности и стереотипы. Для уничтожения объектов, которые создаются на время выполнения своих действий, нужно предусмотреть явное сообщение.

Наиболее простые случаи ветвления процесса взаимодействия можно изобразить на одной диаграмме с использованием соответствующих графических примитивов. Однако следует помнить, что каждый альтернативный поток управления может существенно затруднить понимание построенной модели. Поэтому общим правилом является визуализация каждого потока управления на отдельной диаграмме последовательности. В этой ситуации такие отдельные диаграммы должны рассматриваться совместно как одна модель взаимодействия.

Дальнейшая детализация диаграммы последовательности связана с введением временных ограничений на выполнение отдельных действий в системе. Для простых асинхронных сообщений временные ограничения могут отсутствовать. Однако необходимость синхронизировать сложные потоки управления, как правило, требуют введение в модель таких ограничений. Общая их запись должна следовать семантике языка объектных ограничений, который рассмотрен в приложении.

### **3.4.2. Разработка диаграмм деятельности (activity diagram)**

При моделировании поведения проектируемой или анализируемой системы возникает необходимость не только представить процесс изменения ее состояний, но и детализировать особенности реализации выполняемых системой операций. Традиционно для этой цели использовались блок-схемы или структурные схемы алгоритмов. Каждая такая схема акцентирует внимание на последовательности выполнения определенных действий или элементарных операций, которые в совокупности приводят к получению желаемого результата.

Алгоритмические и логические операции, требующие выполнения в определенной последовательности, окружают нас постоянно. Конечно, мы не всегда задумываемся о том, что подобные операции относятся к столь научным категори-

ям. Например, чтобы позвонить по телефону, нам предварительно нужно снять трубку или включить его. Для приготовления кофе или заваривания чая необходимо вначале вскипятить воду. Чтобы выполнить ремонт двигателя автомобиля, требуется осуществить целый ряд нетривиальных операций, таких как разборка силового агрегата, снятие генератора и некоторых других.

Для моделирования процесса выполнения операций в языке UML используются так называемые диаграммы деятельности. Применяемая в них графическая нотация во многом похожа на нотацию диаграммы состояний, поскольку на диаграммах деятельности также присутствуют обозначения состояний и переходов. Отличие заключается в семантике состояний, которые используются для представления не деятельности, а действий, и в отсутствии на переходах сигнатуры событий. Каждое состояние на диаграмме деятельности соответствует выполнению некоторой элементарной операции, а переход в следующее состояние срабатывает только при завершении этой, операции в предыдущем состоянии. Графически диаграмма деятельности представляется в форме графа деятельности, вершинами которого являются состояния действия, а дугами — переходы от одного состояния действия к другому.

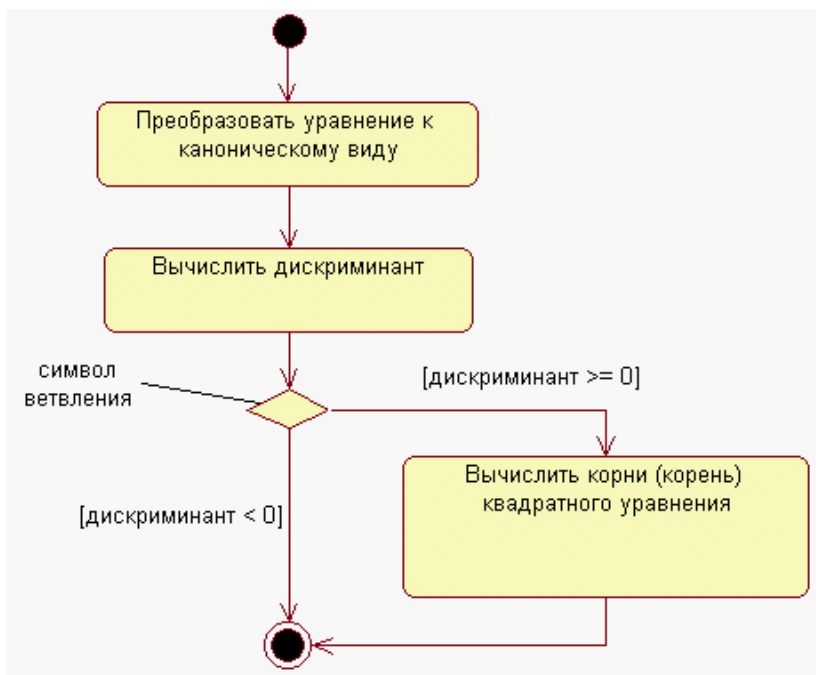


Рис. 17. Фрагмент диаграммы деятельности для алгоритма нахождения корней квадратного уравнения





Рис. 18. Различные варианты ветвлений на диаграмме деятельности

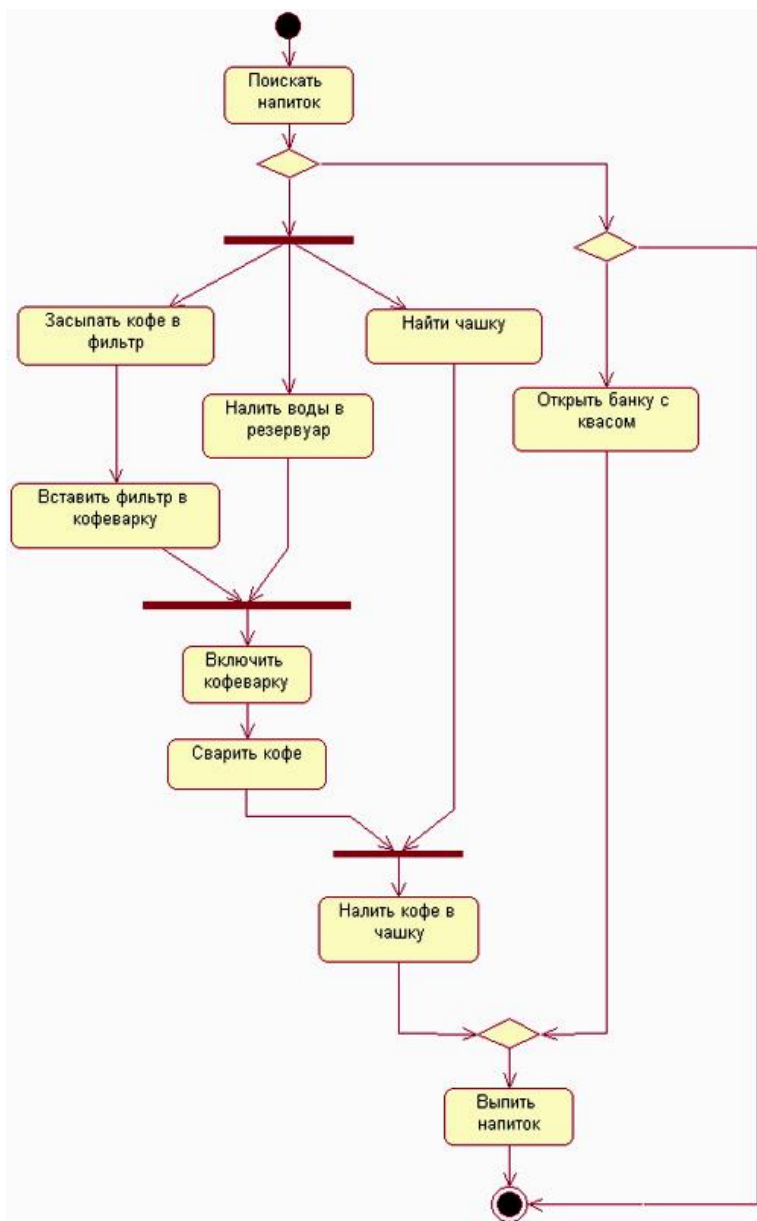


Рис. 19. Диаграмма деятельности для примера с приготовлением напитка

Диаграммы деятельности играют важную роль в понимании процессов реализации алгоритмов выполнения операций классов и потоков управления в моделируемой системе. Используемые для этой цели традиционные блок-схемы алгоритмов обладают серьёзными ограничениями в представлении параллельных процессов и их синхронизации. Применение дорожек и объектов открывает дополнительные возможности для наглядного представления бизнес-процессов, позволяя специфицировать деятельность подразделений компаний и фирм.

Содержание диаграммы деятельности во многом напоминает диаграмму состояний, хотя и не тождественно ей. Поэтому многие рекомендации по построению последней оказываются справедливыми применительно к диаграмме деятельности. В частности, эта диаграмма строится для отдельного класса, варианта использования, отдельной операции класса или целой подсистемы.

С одной стороны, на начальных этапах проектирования, когда детали реализации деятельностей в проектируемой системе неизвестны, построение диаграммы деятельности начинают с выделения под-деятельностей, которые в совокупности образуют деятельность подсистем. В последующем, по мере разработки диаграмм классов и состояний, эти под-деятельности уточняются в виде отдельных вложенных диаграмм деятельности компонентов подсистем, какими выступают классы и объекты.

С другой стороны, отдельные участки рабочего процесса в существующей системе могут быть хорошо отлаженными, и у разработчиков может возникнуть желание сохранить этот механизм выполнения действий в проектируемой системе. Тогда строится диаграмма деятельности для этих участков, отражающая конкретные особенности выполнения действий с использованием дорожек и объектов. В последующем такая диаграмма вкладывается в более общие диаграммы деятельности для подсистемы и системы в целом, сохраняя свой уровень детализации.

Таким образом, процесс объектно-ориентированного анализа и проектирования сложных систем представляется как последовательность итераций нисходящей и восходящей разработки отдельных диаграмм, включая и диаграмму деятельности. Доминирование того или иного из направлений разработки определяется особенностями конкретного проекта и его новизной.

В случае типового проекта большинство деталей реализации действий могут быть известны заранее на основе анализа существующих систем или предшествующего опыта разработки систем-прототипов. Для этой ситуации доминирующим будет восходящий процесс разработки (Зачем изобретать велосипед заново?). Использование типовых решений может существенно сократить время разработки и избежать возможных ошибок при реализации проекта.

При разработке проекта новой системы, процесс функционирования которой основан на новых технологических решениях, ситуация представляется более сложной. А именно, до начала работы над проектом могут быть неизвестны не только детали реализации отдельных деятельностей, но и само содержание этих деятельностей становится предметом разработки. В данном случае доминирующим будет нисходящий процесс разработки от более общих схем к уточняющим их диаграммам. При этом достижение такого уровня детализации всех диаграмм, который достаточен для понимания особенностей реализации всех действий и деятельностей, может служить признаком завершения отдельных этапов работы над проектом.

В заключение следует заметить, что диаграмма деятельности, так же как и другие виды канонических диаграмм, не содержит средств выбора оптимальных решений. При разработке сложных проектов проблема выбора оптимальных решений становится весьма актуальной. Рациональное расходование средств, затраченных на разработку и эксплуатацию системы, повышение ее производительности и надежности зачастую определяют конечный результат всего проекта. В

такой ситуации можно рекомендовать использование дополнительных средств и методов, ориентированных на аналитико-имитационное исследование моделей системы на этапе разработки ее проекта.

В частности, при построении диаграмм деятельности сложных систем могут быть успешно использованы различные классы сетей Петри (классические, логико-алгебраические, стохастические, нечеткие и др.) и нейронных сетей. Применение этих формализмов позволяет не только получить оптимальную структуру поведения системы на ее модели, но и специфицировать целый ряд дополнительных характеристик системы, которые не могут быть представлены на диаграмме деятельности и других диаграммах UML.

## 4. ПЛАН ВЫПОЛНЕНИЯ ЛАБОРАТОРНЫХ РАБОТ И КУРСОВОГО ПРОЕКТА

В рамках выполнения лабораторных работ и курсового проекта необходимо разработать проект информационной системы в соответствии с ГОСТами:

- ГОСТ Р ИСО/МЭК 15288-2005 Информационная технология. Системная инженерия. Процессы жизненного цикла систем (см. ISO/IEC 15288:2008)

- ГОСТ Р ИСО/МЭК 12207-2010 Информационная технология. Системная и программная инженерия. Процессы жизненного цикла программных средств (см. ISO/IEC 12207:2008)

- ГОСТ 34.602-89 Техническое задание на создание автоматизированной системы

- РД 50-34.698-90 Автоматизированные системы; требования к содержанию документов (общее описание системы, описание организационной структуры, руководство пользователя, описание комплекса технических средств и схемы подключения, описание входных и выходных данных, описание ПО, описание математического обеспечения, содержание организационно-распорядительных документов)

Тему для проекта необходимо придумать самостоятельно, но согласовать с преподавателем. Возможно совпадение темы к темой диплома или курсовой работы по курсу «Проектирование ИС». Возможные варианты тем: “Реализация этапов анализа, проектирования, разработки и внедрения информационной системы автоматизации деятельности регистратуры поликлиники”; “Реализация этапов анализа, проектирования, разработки и внедрения информационной системы автоматизации деятельности автосервиса”.

Таким образом, необходимо разработать проект информационной системы, начиная с анализа предметной области и создания ТЗ, и заканчивая вводом в эксплуатацию в соответствии с существующими ГОСТами и описанием 7 этапов, которые описаны ниже. Предполагается, что проектируе-

мая система должна иметь сложную разветвленную многоуровневую архитектуру.

Стадии и этапы создания ИС, выполняемые организациями-участниками, прописываются в договорах и технических заданиях на выполнение работ:

Этап 1. Формирование требований к ИС.

На начальной стадии проектирования выделяют следующие этапы работ:

- обследование объекта и обоснование необходимости создания ИС;

- формирование требований пользователей к ИС;

- оформление отчета о выполненной работе и тактико-технического задания на разработку.

Этап 2. Разработка концепции ИС.

- изучение объекта автоматизации;

- проведение необходимых научно-исследовательских работ;

- разработка вариантов концепции ИС, удовлетворяющих требованиям пользователей;

- оформление отчета и утверждение концепции.

Этап 3. Техническое задание.

- разработка и утверждение технического задания на создание ИС.

Этап 4. Эскизный проект.

- разработка предварительных проектных решений по системе и ее частям;

- разработка эскизной документации на ИС и ее части.

Этап 5. Технический проект.

- разработка проектных решений по системе и ее частям;

- разработка документации на ИС и ее части;

- разработка и оформление документации на поставку комплектующих изделий;

- разработка заданий на проектирование в смежных частях проекта.

Этап 6. Рабочая документация.

- разработка рабочей документации на ИС и ее части;
- разработка и адаптация программ.

Этап 7. Ввод в действие.

- подготовка объекта автоматизации;
- подготовка персонала;
- комплектация ИС поставляемыми изделиями;
- строительно-монтажные работы;
- пусконаладочные работы;
- проведение предварительных испытаний;
- проведение опытной эксплуатации;
- проведение приемочных испытаний.

Конечным итогом обучения студента в рамках курса «Системная инженерия» является курсовой проект, который должен содержать следующие разделы:

1. Анализ предметной области и формирование требований к ИС (не менее 3 страниц, этапы 1,2)
2. Разработка технического задания (не менее 15 страниц, ГОСТ 34.602-89, ГОСТ Р ИСО/МЭК 15288-2005, ISO/IEC 15288:2008, ISO/IEC 12207:2008, этап 3)
3. Разработка эскизного проекта. (этап 4, не менее 5 страниц)
4. Разработка технического проекта (этап 5, не менее 3 страниц);
5. Проектирование ИС, разработка концептуальной и логических моделей (язык UML, диаграмма вариантов использования, диаграмма классов, диаграмма последовательности и диаграмма деятельности)
6. Проектирование ИС, разработка физических моделей (язык UML, диаграмма компонентов и диаграмма развертывания)
7. Реализация основных функций системы (реализация функций, описанных диаграммами последовательности на любом языке программирования)
8. Создание рабочей документации (этап 6, РД 50-34.698-90, включая документацию по основным диаграммам)
9. Ввод в эксплуатацию (этап 7)



## ЗАКЛЮЧЕНИЕ

Системная инженерия на всех этапах от выявления нужд потребителей и до определения необходимых функциональных возможностей системы, от документирования требований и до разработки и реализации проекта и проверки соответствия сосредоточена на всестороннем рассмотрении полного ЖЦ.

Учебное пособие по курсу «Системная инженерия» предназначено для получения студентами целостного представления о системной инженерии, как междисциплинарной области технических наук, сосредоточенной на проблемах разработки и построения сложных, комплексных искусственных систем.

В процессе изучения дисциплины студенты получают теоретические знания о комплексе технических, организационных и управленческих вопросов создания эффективных систем, а также практические навыки по разработке моделей процессов системной инженерии и жизненного цикла систем.

В рамках учебного пособия рассмотрены следующие вопросы:

- система международных и национальных стандартов в области системной и программной инженерии;
- технологии проектирования корпоративных систем управления с использованием программных инструментальных средств;
- технические средства и инструменты реализации процессов системной и программной инженерии.
- основные понятия и концепции системной инженерии;
- базовые методы и средства системной и программной инженерии;
- ГОСТы и международные стандарты в области ИТ;
- ГОСТ 34.601-90 Автоматизированные системы; стадии создания;

- ГОСТ 34.602-89 Техническое задание на создание автоматизированной системы;
- РД 50-34.698-90 Автоматизированные системы; требования к содержанию документов;
- методы управления развитием ИС; основы проектирования ИС (модель жизненного цикла, технологии и методологии разработки);
- основы проектирования ИС: модель жизненного цикла, технологии и методологии разработки;
- порядок планирования и реализации модели жизненного цикла и моделей процессов системной и программной инженерии при создании систем.

Учебное пособие содержит задания на лабораторные работы и курсовое проектирование, предоставляющие возможность получения практических навыков системной инженерии для решения прикладных задач.

## БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Королев, Е.Н. Объектно-ориентированные технологии создания программных продуктов на языке Java[Текст]: учеб. пособие / Е.Н. Королев.- Воронеж: ВГТУ, 2006. - 127 с.

2. Королев, Е.Н. Проектирование и разработка приложений на языке Java[Текст]: учеб. пособие / Е.Н. Королев.- Воронеж: ВГТУ, 2008.- 137 с.

3. Королев, Е.Н. Программирование под Internet на языке Java[Текст]: учеб. пособие / Е.Н. Королев.- Воронеж: ВГТУ, 2003.- 209 с.

## ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	3
1. ПРОЦЕССЫ ЖИЗНЕННОГО ЦИКЛА СИСТЕМЫ	8
1.1 Процессы соглашения	9
1.2 Процессы предприятия	14
1.3 Процессы проекта	22
1.4 Технические процессы	35
2. СТАДИИ ЖИЗНЕННОГО ЦИКЛА СИСТЕМЫ	54
2.1 Стадия замысла	54
2.2 Стадия разработки	56
2.3 Стадия производства	59
2.4 Стадия применения	60
2.5 Стадия поддержки применения	62
2.6 Стадия изъятия и списания	64
3. ПОСТРОЕНИЕ ИНТЕГРИРОВАННОЙ МОДЕЛИ СЛОЖНОЙ СИСТЕМЫ	66
3.1. Язык моделирования UML	66
3.2. Построение концептуальной модели.	73
3.2.1. Разработка диаграмм вариантов использования	73
3.2.2. Пример построения диаграммы вариантов использования	77
3.2.3. Рекомендации по разработке диаграмм вариантов использования	80
3.3. Построение логической модели, отражающей статические аспекты работы сложной системы	84
3.4. Построение логической модели, отражающей динамические аспекты работы сложной системы	96
3.4.1. Разработка диаграмм последовательности (sequence diagram)	96
3.4.2. Разработка диаграмм деятельности (activity diagram)	110
4. ПЛАН ВЫПОЛНЕНИЯ ЛАБОРАТОРНЫХ РАБОТ И КУРСОВОГО ПРОЕКТА	118
ЗАКЛЮЧЕНИЕ	121
БИБЛИОГРАФИЧЕСКИЙ СПИСОК	123

Учебное издание

Королев Евгений Николаевич

## МЕТОДЫ СИСТЕМНОЙ ИНЖЕНЕРИИ

В авторской редакции

Компьютерная верстка Е.Н. Королев

Подписано к изданию 14.06.2016.

*Объем данных 1,6 Мб*

ФГБОУ ВО «Воронежский государственный технический  
университет»

394026 Воронеж, Московский просп., 14