Министерство образования и науки Республики Казахстан

Карагандинский университет имени академика Е.А. Букетова

Факультет Математики и информационных технологий

Кафедра Прикладной математики и информатики

**Спирина Елена Александровна**

**Курс лекций**

**по дисциплине «Управление IT-проектами»**

Образовательная программа: «6B06103-Информационные системы»»

Караганда 2023

Лекция 1. Понятие и основные элементы ИТ-инфраструктуры предприятия

*План:*

1. ИТ-инфраструктура предприятия
2. Базовый подход к построению архитектуры предприятия
3. ИТ-стратегия предприятия
4. Основные Компоненты ИТ-инфраструктуры
5. Современные подходы к управлению производственными процессами

**1. ИТ-инфраструктура предприятия**

**IT-инфраструктура** — это совокупность компонентов, необходимых для работы и управления корпоративными средами. Компоненты ИТ-инфраструктуры включают в себя: хранилище данных, операционную систему, сетевые ресурсы, программное и аппаратное обеспечение.

**IT-инфраструктура** – это комплексная система, объединяющая в себе все аппаратное и программное обеспечение для бесперебойного функционирования информационного пространства предприятия либо компании. Доступная, надежная и безопасная ИТ-инфраструктура предприятия дает существенное конкурентное преимущество на рынке, повышает производительность сервисов и позволяет достичь поставленных целей.

**Эволюция ИТ-инфраструктуры.**

В эволюции ИТ-инфраструктуры можно выделить пять этапов, обозначенные на рисунке 1.1.

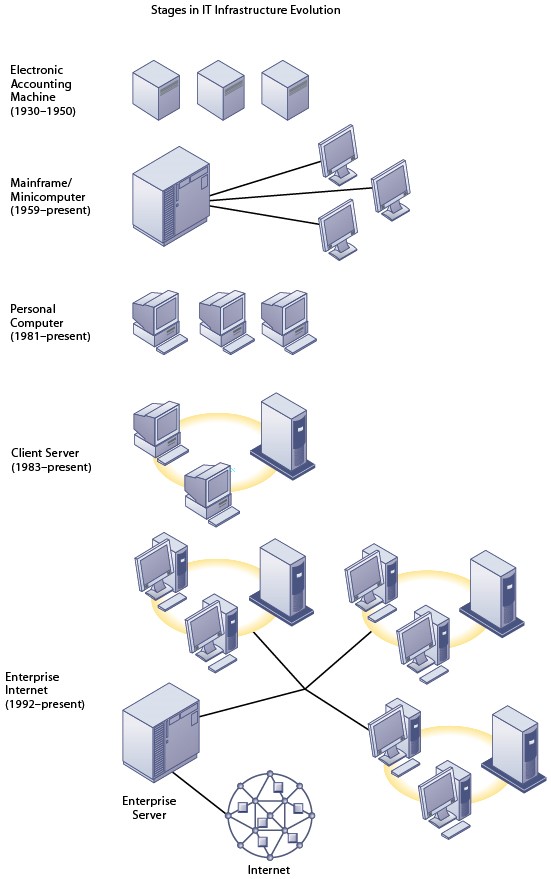


Рисунок 1.1. Эпохи в революции ИТ-инфраструктуры

1**.   Эпоха электронных учетных машин (1930-1950):** использование больших, громоздких машин с встроенным программным обеспечением для сортировки, добавления и представления данных.

2.    **Эра мэйнфреймов общего назначения и миникомпьютеров (1959- настоящее время):** внедрение и дальнейшее использование мэйнфреймов. Мэйнфреймы были первыми мощными компьютерами, которые могли обеспечивать совместное использование времени, многозадачность и виртуальную память, и стали достаточно мощными, чтобы поддерживать тысячи удаленных терминалов. Эпоха мэйнфреймов была периодом высокоцентрализованных вычислений, управляемых программистами и системными операторами. Миникомпьютеры, мощные, но менее дорогие компьютеры, начали менять эту модель, позволяя децентрализовать вычисления, настраиваемые для отдельных отделов или бизнес-единиц.

3.  **Эра персональных компьютеров (1981-настоящее время)**: Появление IBM PC в 1981 году обычно считается началом эры ПК, потому что эта машина была первой, получившей широкое распространение в американском бизнесе. 95 процентов из 1 миллиарда современных компьютеров - это компьютеры Wintel, использующие программное обеспечение Windows и микропроцессоры Intel. ПК были автономными системами, пока программное обеспечение операционной системы ПК в 1990-х годах не позволило связать их в сети.

4.   **Эпоха клиент-сервера (1983- настоящее время**): В клиент-серверных вычислениях настольные или портативные компьютеры, называемые клиентами, подключаются к серверным компьютерам, которые предоставляют клиентам услуги и возможности. Работа по компьютерной обработке разделена между этими двумя типами машин. Клиент является точкой входа пользователя, в то время как сервер обычно обрабатывает и хранит общие данные, обслуживает веб-страницы или управляет сетевыми действиями. Термин «сервер» относится как к программному приложению, так и к физическому компьютеру, на котором работает сетевое программное обеспечение. Сервер может быть мэйнфреймом, но сегодня серверные компьютеры обычно представляют собой более мощные версии персональных компьютеров.

В двухуровневой клиент-серверной архитектуре клиентский компьютер подключен к серверу по сети с разделением обработки между ними. В многоуровневой (N-уровневой) клиент-серверной архитектуре работа всей сети сбалансирована на нескольких разных уровнях серверов. Распределение работы между несколькими небольшими недорогими машинами обходится намного дешевле, чем миникомпьютеры или мэйнфреймы.

Визуально 4 этап эволюции представлен на рисунке 1.2.

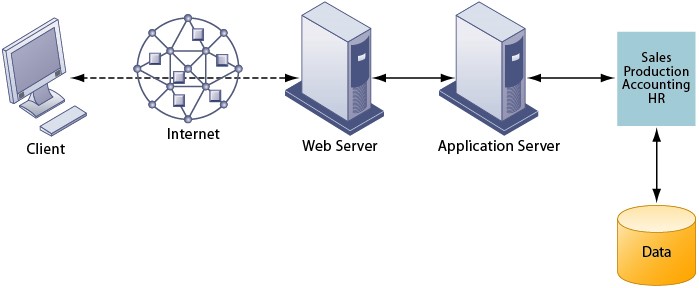


Рисунок 1.2. Многоуровневая сеть клиент/сервер (N-уровневая)

В многоуровневой клиент-серверной сети клиентские запросы на обслуживание обрабатываются серверами разных уровней.

**5. Корпоративный Интернет (1992-настоящее время):** Пакет протоколов управления передачей сетевых технологий Интернета/Internet Protocol (TCP/IP) позволяет предприятиям объединять разрозненные устройства и локальные вычислительные сети (LAN) в единую общеорганизационную сеть. Интегрированные вычислительные среды обеспечивают гораздо более быстрый и бесперебойный сбор и распределение данных.

**ИТ-инфраструктура компании** – это основа для обслуживания клиентов, работы с поставщиками и управления бизнес-процессами. Она определяет возможности компании сегодня и ее возможности в будущем.

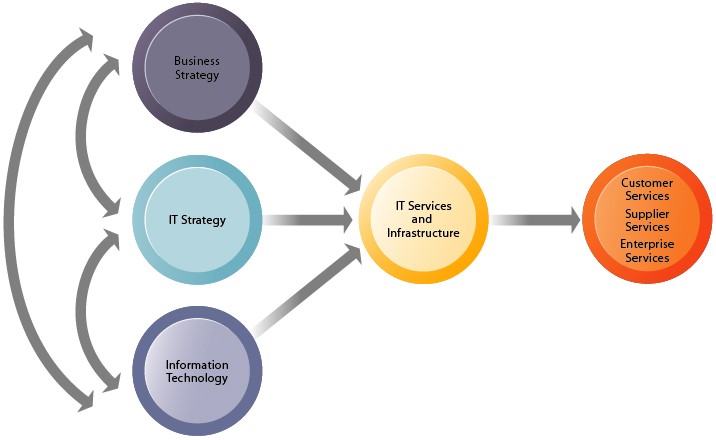


Рисунок 1.3. Связь между компанией, ИТ-инфраструктурой и бизнес-возможностями.

Услуги, которые компания предоставляет своим клиентам, поставщикам и сотрудникам, являются прямой функцией ее ИТ-инфраструктуры. В идеале эта инфраструктура должна поддерживать бизнес-стратегию фирмы и стратегию информационных систем. Новые информационные технологии оказывают мощное влияние на бизнес и ИТ-стратегии, а также на услуги, которые могут быть предоставлены клиентам.

ИТ-инфраструктуру можно рассматривать как технологические или сервисные кластеры. Определение, основанное на услугах, фокусируется на услугах, предоставляемых аппаратным и программным обеспечением, таких как:

·         вычислительные платформы;

·         телекоммуникации;

·         управление физическими объектами;

·         прикладное программное обеспечение;

·         управление данными;

·         управление ИТ;

·         ИТ-стандарты;

·         ИТ-образование и ИТ-исследования и разработки.

Перспектива сервисной платформы подчеркивает ценность для бизнеса, обеспечиваемую ИТ-инфраструктурой.

Развитие рынка заставляет организацию менять модели бизнеса, что, в свою очередь требует адекватных изменений ИТ-инфраструктуры.

*Оптимизировать управление ИТ-инфраструктурой возможно на основе современных концепций*:

- управ­ления ИТ-подразделениями как поставщиками услуг для бизнеса;

- управление уровнем ИТ-сервисов для бизнеса;

- управление бизнес-сервисами.

Современные информационные технологии (ИТ) становятся неотъемлемой составляющей любого предприятия. Сегодня они для многих предприятий - не просто способ автоматизации рутинныхопераций (технологическая подложка), а эффективный инструмент в конкурентной борьбе.

Современные ИТ–системы призваны быстро адаптироваться к новым потребностям бизнеса (его целям задачам) и полностью *соответствовать архитектуре предприятия (Enterprise Architecture EA).*

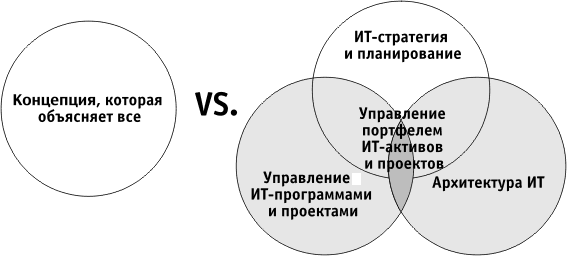
Под **архитектурой предприятия (EA - Enterprise Architecture)**  *понимается* ***полное описание (модель) структуры предприятия, как системы, включающее описание ключевых элементов этой системы, связей между ними*.**

*Архитектура предприятия определяет общую структуру и функции систем* (бизнес и ИТ) в рамках всей организации в целом (включая партнеров и другие организации, формирующие так называемое «предприятие реального времени») и обеспечивает общую рамочную модель (framework), стандарты и руководства для архитектуры уровня отдельных проектов. Общее видение, обеспечиваемое архитектурой предприятия, создает возможность единого **проектирования систем, адекватных, с точки зрения обеспечения потребностей организации**, и способных к взаимодействию и интеграции там, где это необходимо.

Разработка стратегии современного предприятия (Strategy and Planning) и управление корпоративными проектами (Enterprise program management) включают в себя направление, связанное непосредственно с информационными технологиями.

Современные тенденции рассматривают *ИТ проекты и стратегические инициативы как определенный актив компании, которым можно управлять аналогично финансовым активам*.

**Взаимосвязь бизнеса и ИТ:**



(Данилин А., Слюсаренко А. Архитектура и стратегия: «Инь» и «Янь» информационных технологий предприятия.)

Рисунок 1.4. Взаимосвязь бизнеса и ИТ

**Управление портфелем информационных технологий** (Business and IT portfolio management) – это процесс управления инвестициями в области управления ИТ проектами. Под портфелем понимается совокупность проектов, выполняемых на общем пуле ресурсов (финансы, люди, оборудование, материалы, энергия), при этом пул ресурсов и результаты всех проектов портфеля находятся в компетенции одного центра ответственности. Аналитики компании META Group считают, что это - область пересечения архитектуры предприятия, стратегии предприятия и управления корпоративными проектами (рисунок 1.5).

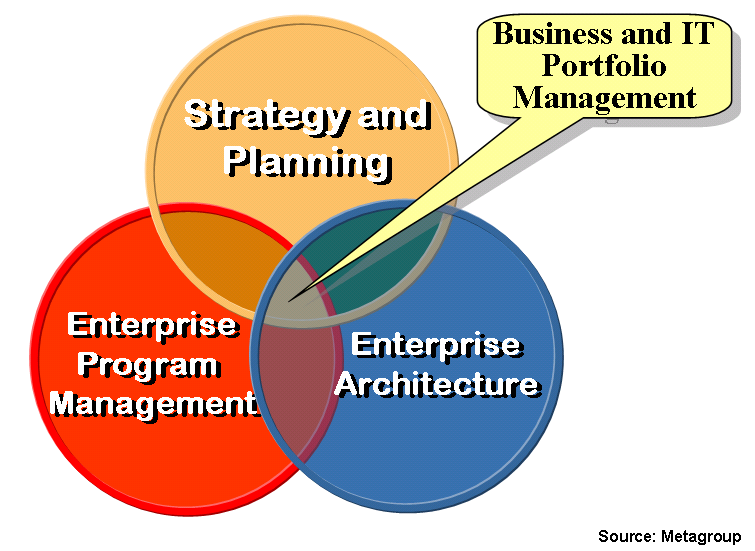


Рисунок 1.5. Управление портфелем ИТ

Стратегия и планирование при этом обеспечивают основу для выработки ИТ стратегии предприятия, в соответствии с которыми появляются *проекты внедрения (модернизации) информационных систем*.

***Управление проектами*** – можно рассматривать, в первую очередь, как механизм, обеспечивающий переход от текущего состояния к планируемому, или, другими словами, переход от текущей архитектуры предприятия к целевой архитектуре.

Представление информационных технологий в виде активов, позволяет предприятию корректно оценивать и расставлять приоритеты при вложении инвестиций и управлении ИТ проектами (активами) с учетом приемлемого уровня риска, и, таким образом планировать инвестиции в эту область. Считается, что **управление ИТ портфелем** должно преследовать три основные цели:

* максимизация ценности портфеля;
* синхронизация ИТ - портфеля с требованиями бизнеса;
* поиск оптимального баланса между риском и потенциальной отдачей от ИТ - портфеля.

**Архитектура предприятия является одним из элементов управление ИТ портфелем и предоставляет необходимую информацию о бизнес-процессах и технологиях, необходимых для их автоматизации**. Архитектура предприятия не только является основой для разработки портфеля активов, но также обеспечивает весь жизненный цикл многих ИТ - активов.

Архитектура предприятия позволяет увидеть все предприятие целиком. Создать цепочку, показывающую воздействие отдельных элементов стратегии развития предприятия на его бизнес-процессы, и их зависимость от информационных систем и технологических элементов.

Любому предприятию требуется планомерное развитие его структуры, бизнес-процессов, информационных систем и их интеграция между собой.

**Архитектура предприятия собственно и является планом развития предприятия** **(целевая архитектура) и документированной схемой того, что происходит в компании в текущий момент времени** (**текущая архитектура**).

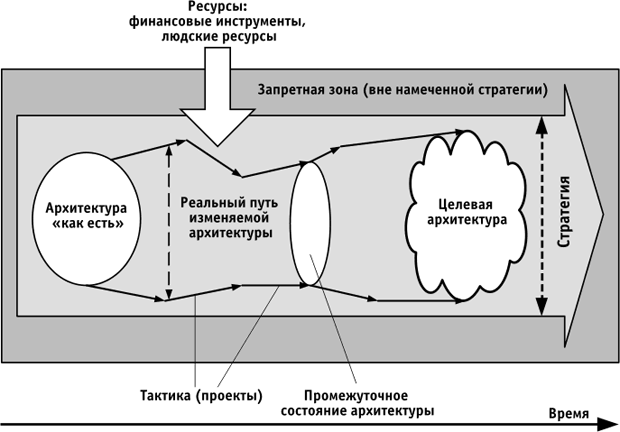


Рисунок 1.6

**Текущая архитектура (Current architecture)** - описывает существующее состояние архитектуры предприятия. Называется также архитектурой “как есть” (AS-IS) или базовым состоянием существующей архитектуры.

Текущая архитектура – это отображение объективной реальности, включающей в себя существующие компоненты (бизнес-процессы, информационные системы, технологические элементы) и их связи. Это набор моделей с неизбежными упрощениями, ограничениями и субъективными искажениями.

**Процесс разработки текущей архитектуры – это, в первую очередь, процесс документирования и поддержания информации о состоянии предприятия в актуальном виде,** обеспечивающий регистрацию и контроль информации обо всех элементах архитектуры предприятия, включающий в себя ведение базы данных по архитектурным объектам; ведение управленческого учета и учета состояния.

Процесс разработки текущей архитектуры аналогичен процессу ITIL/ITSM (управление конфигурацией - Configuration Management). Для упрощения работы по разработке текущей архитектуры многие компании используют базу данных конфигурационных единиц (CMDB), дополнив ее необходимой информацией.

**Целевая архитектура (Target Architecture) -** описывает желаемое будущее состояние предприятия или "что должно быть сформировано" (TO-BE). Другими словами, целевая архитектура является будущей моделью предприятия.

Целевую архитектуру можно назвать идеальной моделью предприятия, в основу которой заложены:

* стратегические требования к бизнес-процессам и информационным технологиям;
* информация о выявленных «узких местах» и путях их устранения;
* анализ технологических тенденций и среды бизнес деятельности предприятия.

**Целевая архитектура (модель to-be) и текущая архитектура (модель as-is)** позволяют описать начальное и конечное состояние предприятия – до и после внесения изменений в его структуру, оставляя без внимания сам процесс изменений.

*Процесс перехода от текущей архитектуры предприятия к целевой переводит предприятие на новую спираль развития и, таким образом, мы можем говорить, что архитектура предприятия характеризуется определенным жизненным циклом, похожим на жизненный цикл информационных систем.*

**2. Базовый подход к построению архитектуры предприятия**

**Современные подходы к построению архитектуры предприятия традиционно разделяют ее на несколько слоев** (предметных областей). Количество архитектурных слоев варьируется в различных методиках. Ниже мы рассмотрим слои, использующиеся в большинстве из существующих методик (рисунок 1.7):

1. Стратегические цели и задачи предприятия.
2. Бизнес – архитектура предприятия.
3. Архитектура информационных технологий (ИТ архитектура предприятия).

***Архитектуру информационных технологий, в свою очередь, разделяют на:***

3.1 Информационную архитектуру (Enterprise Information Architecture - EIA).

3.2 Архитектуру прикладных решений (Enterprise Solution Architecture - ESA).

3.3 Технологическую архитектуру (Enterprise Technical Architecture - ETA).



Рисунок 1.7. Основные слои архитектуры предприятия

**1. Стратегические цели и задачи предприятия** определяют основные направления развития и ставят долгосрочные задачи и цели. При разработке стратегических целей предприятия необходимо учитывать воздействие информационных технологий на формирование облика современного предприятия.

В ходе разработки стратегических целей предприятия формируется (модернизируется) и стратегия развития информационных технологий.

**Бизнес стратегия** – определяет направление развития бизнеса в соответствии со стратегическими целями и задачами, стоящими перед предприятием, и отвечает на вопрос, **почему предприятие должно развиваться именно в этом направлении**. Бизнес стратегия включает в себя:

* Цели и задачи стоящие перед предприятием.
* Бизнес решения, необходимые для достижения поставленных целей и задач.
* Изменения, которые нужно провести для достижения поставленных целей и задач.

## **2. Бизнес – архитектура предприятия**

**Бизнес - архитектура предприятия (EBA - Enterprise Business Architecture)** – это целевое построение организационной структуры предприятия, увязанное с его миссией, стратегией, бизнес - целями.

В ходе построения бизнес - архитектуры определяются необходимые бизнес-процессы, информационные и материальные потоки, а также организационно-штатная структура.

Бизнес - архитектура предприятия неразрывна, связана с процессом его управления. Под управлением предприятием обычно понимается деятельность компании с учетом изменений в окружающей экономической и социальной среде. Управленческий персонал распределяет финансовые, трудовые и материальные ресурсы для максимально эффективного достижения стратегических целей и задач предприятия.

В ходе разработки бизнес - архитектуры подробно рассматриваются различные модели построения предприятия, соответствующие стратегии *его* развития. Модели бизнес - архитектуры могут быть разделены на три класса: классические (эталонные), специализированные и специфические.

* Классическая или, другими словами, эталонная архитектура предприятия является идеальной моделью построения организации.
* Специализированная архитектура – включает в себя модели, ориентированные на предприятия определенных отраслей или определенные фазы производства. В основе специализированных методологических моделей, как правило, лежат исторически сложившиеся алгоритмы управления в данных отраслях (например, банки, химическая промышленность, телекоммуникации).
* Специфическая архитектура - так обычноназывают исторически сложившуюся на данном предприятии модель бизнес - процессов.

**Построение бизнес - архитектуры начинается с описания контекста бизнес - архитектуры**.

Общее видение бизнес - архитектуры предприятия включает анализ основных функций, цепочек создания добавленной стоимости (Value Landscape Analysis), модели бизнес – сценариев (Business Scenario Models), анализ информационных связей и процессов (Information Value Chain Analysis).

**Основу архитектуры предприятия** составляет:

**-** анализ бизнес событий (Business Event Analysis),

**-** декомпозиция функций и процессов (Function / Process Decomposition),

**-** модель расположения (Location Model),

**-** модель интеграции (Integration Model).

***Бизнес - архитектура представляется в виде набора бизнес моделей.***

**Бизнес модели** – это «набор событий, связанных с бизнесом, в который вовлечены различные функции бизнеса, организационные единицы и активы предприятия».

В настоящее время существуют различные методики описания бизнес - архитектуры предприятия.

**Основу бизнес архитектуры предприятия составляют модели бизнес-процессов.**

* функциональные модели,
* организационные модели,
* модели процессов/потоков работ, модели данных/ресурсов,
* модели причинно-следственных связей.

**Декомпозиция бизнес-процессов** – методика, описания бизнес-процессов в виде последовательной их детализации.

Декомпозиция - это процесс создания диаграммы, детализирующей определенный блок и связанные с ним дуги.

*Декомпозиция бизнес-процессов обеспечивает их последовательную детализацию, определение границ основных организационных единиц. Декомпозиция позволяет определить вклад каждого из них в цепочку добавленной стоимости.*

***В ходе проведения декомпозиции бизнес процессов*** необходимо выполнить следующие шаги:

* определить границы анализа за счет рассмотрения основных функций предприятия;
* выделить ключевые бизнес-процессы;
* выделить дублирующие бизнес-процессы и точки их пересечения.

**Анализ бизнес - событий** позволяет построить зависимость бизнес-процессов и бизнес – событий, понять какие события, что инициируют. Анализ бизнес - событий позволяет перейти к анализу данных, используемых предприятием.

**Модель местоположения** описывает географическое расположение выполняющихся бизнес функций. Модель местоположения позволяет провести визуализацию организационных единиц и определение мест выполнения бизнес-процессов.

**Модель интеграции** определяет связь бизнес-процессов и бизнес- событий.

**Для моделирования бизнес-процессов используются различные пакеты:**

**1. Bizagi Process Modeler** <http://www.bizagi.com/>

**2. Intalio BPMS**[http://bpms.intalio.com/help.html](http://bpms.intalio.com/help.html%C2%A0)

**3. ARIS Express**[http://www.ariscommunity.com/aris-express](http://www.ariscommunity.com/aris-express%C2%A0)

**4. Camunda** <https://camunda.com/products/modeler/>

**5. AllFusion Process Modeler** <http://www.ca.com/ru/default.aspx>

**6. IBM WebSphere Business Modeler** <http://www-03.ibm.com/software/products/us/en/modeler>

**7. ELMA** <http://www.elma-bpm.ru/>

**8. Comindware Business Application Platform** <https://www.comindware.com/ru/platform/>

**3. Архитектура ИТ или ИТ - архитектура предприятия** определяет правила формирования всех компонентов ИТ, взаимосвязи между ними и бизнес - архитектурой предприятия:

3.1 Enterprise Information Architecture (EIA) – информационная архитектура.

3.2 Enterprise Solution Architecture (ESA) – архитектура прикладных решений.

3.3 Enterprise Technical Architecture (ETA) – техническая архитектура

***3.1 Информационная архитектура*** **(EIA - Enterprise Information Architecture)** или, другими словами, архитектура информации – это (с точки зрения аналитиков компании Meta Group) управляемый набор методик, описывающий информационную модель предприятия и включающий в себя:

* Базы данных и хранилища данных.
* Информационные потоки (как внутри организации, так и связи с внешним миром).

Информационную архитектуру предприятия условно можно назвать уровнем потоков данных. Но, при построении информационной архитектуры предприятия нет необходимости создавать модели всех видов данных, используемых на предприятии. Достаточно обеспечить выбор наиболее важных (критичных для предприятия) данных и моделировать их на высоком уровне абстракции.

***3.2 Архитектура прикладных решений*** **(ESA - Enterprise Solution Architecture)** – или, другими словами, архитектура приложений, включает в себя *совокупность программных продуктов и интерфейсов между ними.*

Архитектуру прикладных решений разделают на два направления:

* Область разработки прикладных систем.
* Портфель прикладных систем.

***Область разработки прикладных систем*** описывает технологическую часть архитектуры прикладных решений и включает в себя:

- программные продукты;

- модели данных;

- интерфейсы (API);

- пользовательские интерфейсы.

Область разработки прикладных систем является техническим описанием конкретных приложений. Соответственно, информацию о данных модулях проще всего представить в виде двух следующих схем:

* Компоненты и структура системы – внутренняя структура системы, включающая в себя информацию о программных модулях и базах данных.
* Взаимодействие с другими системами (интерфейсы) – описывает взаимодействие приложения с внешними объектами (программными продуктами, пользователями).

**Портфель прикладных систем** описывает потребности бизнес-процессов предприятия в информационных технологиях и включает в себя набор интегрированных информационных систем

Архитектура прикладных решений описывает ситуацию, сложившуюся в ИТ - подразделении на текущий момент времени (т.е. это картина, демонстрирующая «технологическое обеспечение» бизнес-процессов, где каждой основной бизнес-функции соответствуют определенные приложения). На основе архитектуры прикладных решений строятся планы последующего развития информационных технологий в компании, разрабатываются планы мероприятий и проектов, необходимых для достижения стратегических целей.

***3.3 Техническая архитектура предприятия*** **(ETA - Enterprise Technical Architecture) – это совокупность программно-аппаратных средств, методов и стандартов, обеспечивающих эффективное функционирование приложений.** Другими словами, под технической архитектурой мы будем понимать *полное описание инфраструктуры предприятия*, включающее в себя:

* Информацию об инфраструктуре предприятия.
* Системное программное обеспечение (СУБД, системы интеграции).
* Стандарты на программно-аппаратные средства.
* Средства обеспечения безопасности (программно-аппаратные).
* Системы управления инфраструктурой.

Техническую архитектуру предприятия можно визуально представить в виде совокупности архитектурных схем приложений, используемых на предприятии.

Визуально техническую архитектуру приложения, в свою очередь, можно представить в виде схемы включающей в себя информацию о серверах, сегментах СКС, компонентах системы, стандартах (использующихся в данном приложении) и взаимосвязях между ними.

Таким образом, под **ИТ-инфраструктурой предприятия** понимается различное множество существующих в ней сервисов и систем, сетей, технических и программных средств, данных, автоматизированных процессов.

Между различными частями ИТ-инфраструктуры существуют всевозможные взаимосвязи: один процесс может быть обеспечен несколькими автоматизированными системами, при этом системы могут обмениваться между собой данными, системы более низкого уровня могут послужить механизмами реализации для систем более высокого уровня и т.п.

**4. ИТ-стратегия предприятия**

В случае ИТ-инфраструктуры на место технического задания и технического проекта приходит **ИТ-стратегия**.

**ИТ-стратегия** ***определяет направление развития информационных технологий в соответствии с целями, задачами и бизнес стратегией предприятия, и определяет, как может быть реализована бизнес стратегия.***

ИТ стратегия включает:

* Проекты, которые можно запустить для выполнения бизнес стратегии.
* Варианты решения текущих задач и проблем.
* Технологии, которые можно использовать для достижения поставленных целей.

**Бизнес стратегия** – определяет направление развития бизнеса в соответствии со стратегическими целями и задачами, стоящими перед предприятием, и отвечает на вопрос, **почему** предприятие должно развиваться именно в этом направлении. Бизнес стратегия включает в себя:

* Цели и задачи стоящие перед предприятием.
* Бизнес решения, необходимые для достижения поставленных целей и задач.

Изменения, которые нужно провести для достижения поставленных целей и задач.

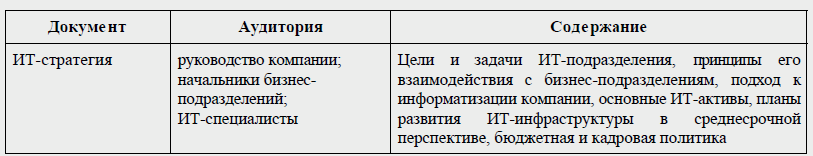
**Архитектура предприятия** строится на основании стратегии (как бизнес, так и ИТ) и определяет, **что** именно требуется сделать для достижения поставленных целей. Архитектура предприятия включает в себя:

* Потребности бизнеса в новых продуктах и услугах.
* Бизнес процессы, обеспечивающие функционирование на текущий момент времени.
* Прикладные системы, автоматизирующие выполнение бизнес процессов.

*ИТ-стратегия выступает в качестве системы приоритетов, правил и планов, которые позволяют достигать адекватности ИТ-инфраструктуры надобностям бизнеса.* Реализация ИТ-стратегии требует владения всевозможной информацией о ней. Такая информация необходима всем представителям предприятия: ИТ-директору и его подчиненным, руководителям бизнес-подразделений, руководству организации, внешним исполнителям, пользователям, консультантам и т.п. Ориентировочный состав технической документации на ИТ-инфраструктуру предприятия представлен в таблице 1.

Таблица 1

Состав технической документации на ИТ-инфраструктуру предприятия





**4. Основные Компоненты ИТ-инфраструктуры**

Вся экосистема состоит из программных решений, различного оборудования и сетевых подключений, которые работают одновременно и дополняют друг друга.

Вся идея заключается в улучшении связи между различными устройствами, будь то настольные компьютеры, планшеты, сканеры, различные серверы и облачные хранилища.

Современная организация любого размера всегда будет опираться на три важнейших ИТ-строительных блока. Их можно разделить на дополнительные подкомпоненты, как описано ниже:

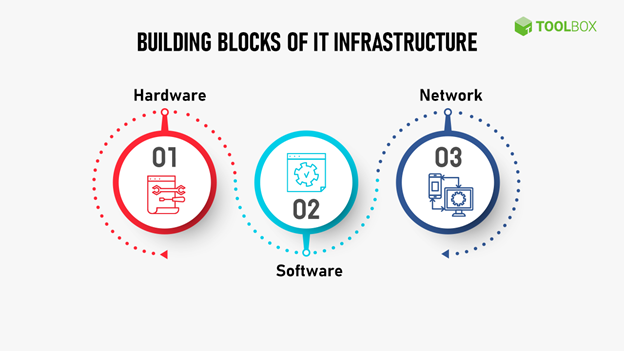


Рисунок 1.9. Строительные блоки ИТ-инфраструктуры

**Основные компоненты ИТ-инфраструктуры (**Рисунок 1.10**)**



Рисунок 1.10.

**Аппаратное обеспечение**

Аппаратное обеспечение относится к физическим компонентам и устройствам, которые помогают вам организовать инфраструктуру. Они являются ее основой. Аппаратное обеспечение относится к таким объектам как:

o   Настольные компьютеры;

o   Ноутбуки;

o   Планшеты, смартфоны и другие мобильные устройства;

o   Серверы и центры обработки данных.

**Программное обеспечение**

Программное обеспечение может включать в себя различные программы и приложения, которые бизнес использует для функционирования, предоставления услуг, управления внутренними конвейерами и многого другого. Кроме того, различные операционные системы могут быть назначены программному обеспечению, поверх которого установлены все программы и приложения.

Итак, программные части включают:

o   Системы управления контентом (CMS);

o   Системы визуализации;

o   Управление взаимоотношениями с клиентами (CRM);

o   Планирование ресурсов предприятия (ERP);

o   Операционные системы;

o   Веб-серверы;

o   Настраиваемое программное обеспечение для внутренней работы.

**Сети**

Сети позволяют объединять устройства в единую сеть и подключать их к Интернету. Соединение защищено брандмауэрами безопасности, которые защищают его от вредоносных программ и взломов. Сеть включает в себя следующие компоненты:

o   Серверы;

o   Центры обработки данных/ Каналы передачи данных;

o   Концентраторы;

o   Коммутаторы;

o   Интернет-концентраторы и маршрутизаторы.

Схематически вариант современной ИТ-инфраструктуры (технологический слой) – рисунок 1.11.

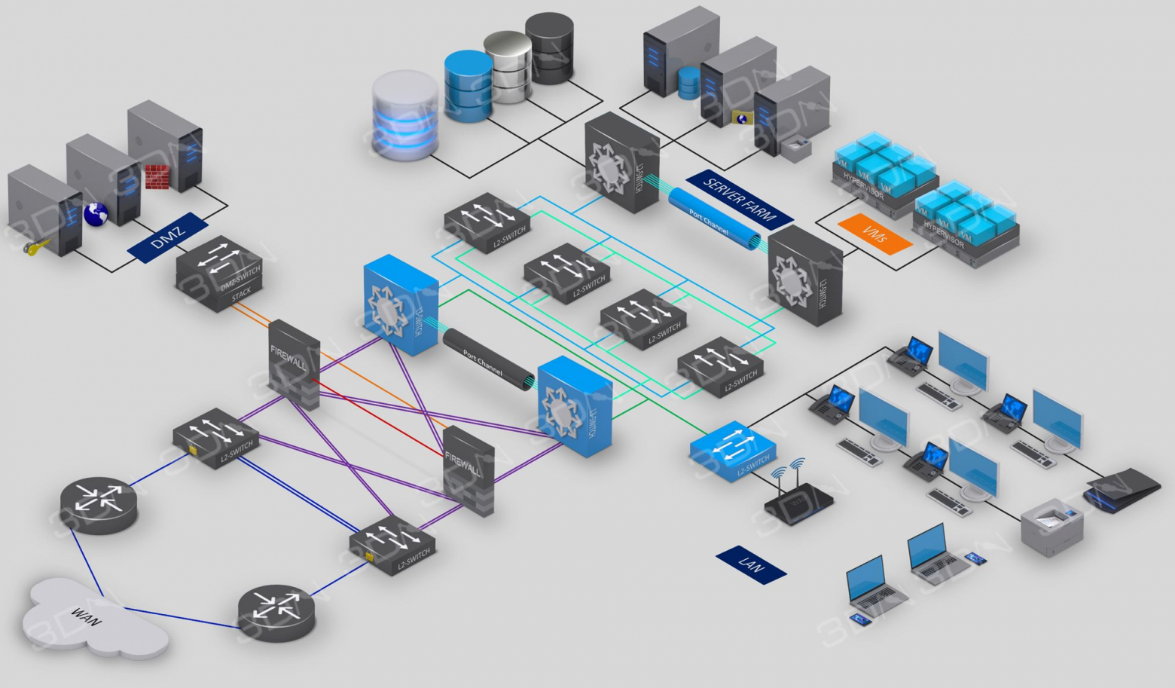


Рисунок 1.11. Принцип организации современной ИТ-инфраструктуры (технологический слой)

Технические ресурсы можно разделить на шесть категорий:

• ресурсы инженерной инфраструктуры;

• вычислительные ресурсы;

• ресурсы передачи данных;

• ресурсы защиты информации;

• ресурсы хранения данных;

• ресурсы коммуникации (ввода-вывода информации с использованием периферийных устройств).

К ИТ-инфраструктуре обычно применяются следующие общие требования:

• предоставлять технические ресурсы с согласованным уровнем качества для решения текущих задач бизнеса;

• быть готовой предоставить в согласованном объеме и в предсказуемые сроки технические ресурсы для решения перспективных задач бизнеса;

• обеспечивать требуемый уровень операционной непрерывности;

• обеспечивать требуемый уровень информационной безопасности;

• соответствовать законодательным и нормативным требованиям.

Схематически взаимосвязь пластов ИТ-инфраструктуры большинства современных предприятий можно обозначить следующим образом (рисунок 1.12).



Рисунок 1.12. Взаимосвязь пластов ИТ-инфраструктуры

**Модели IT-инфраструктуры**

**Традиционная модель инфраструктуры**

Основными компонентами традиционной инфраструктуры являются аппаратное и программное обеспечение. Например, это могут быть различные серверы и настольные ПК. Все оборудование расположено под одной крышей, что упрощает доступ к необходимой информации и вычислительным мощностям. Это может быть сервер под столом или мини-ЦОД (мини-серверная) в подвале офиса.

Эта инфраструктура требует много свободного места на месте развертывания, а также мощности самого оборудования. В результате разработка и установка инфраструктуры такого типа обходятся дороже.

Кроме того, такая система требует постоянного мониторинга, обслуживания и обновления, чтобы соответствовать постоянно меняющимся реалиям рынка.

Это означает, что в дополнение к расходам на оборудование вы также потратите деньги на команду управления инфраструктурой, которая будет следить за ее состоянием, устранять неполадки и своевременно проводить диагностику и модернизацию.

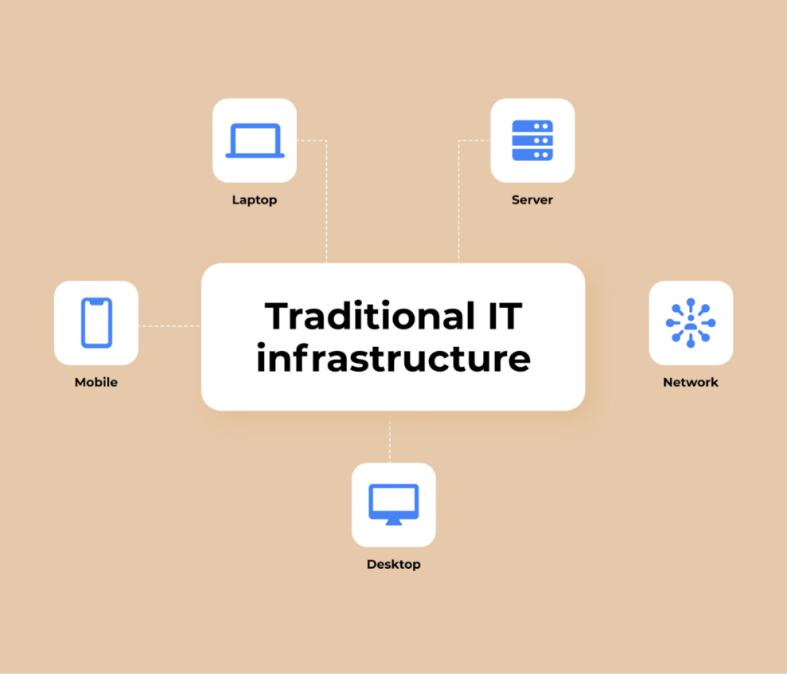


Рисунок 1.13. Традиционная ИТ-инфраструктура

Этот вариант является самым безопасным решением для хранения данных, поскольку позволяет полностью управлять всеми данными и программным обеспечением на предприятии.

***Преимущества традиционной инфраструктуры***

* Вместе с системой вы получаете специализированную команду, которая полностью вовлечена в управление инфраструктурой.
* Наряду с обновлением инфраструктуры вам также необходимо обновить программное обеспечение, что позволяет вам всегда быть на переднем крае новых технологических решений.
* У вас есть возможность работать с различными типами программного обеспечения.

**Облачная модель инфраструктуры**

В наши дни облачные инфраструктуры используются все большим количеством предприятий благодаря их гибкости.

В этом случае компоненты IT-инфраструктуры расположены у облачного провайдера. Все серверы, программные решения и хранилища данных расположены в облаке. Поставщик услуг предоставляет полное техническое обеспечение инфраструктуры, в том числе бесперебойную работу, а клиент управляет ею удаленно — через панель управления и консоль.

В результате вам не придется тратить кучу денег на оборудование и центры обработки данных, которые могут обслуживать внутреннюю ИТ-инфраструктуру вашего предприятия. Вместо этого вы можете найти поставщика облачных услуг и арендовать облачное хранилище, необходимое для переноса в него всего вашего внутреннего конвейера (включая ценные данные, приложения и многое другое).

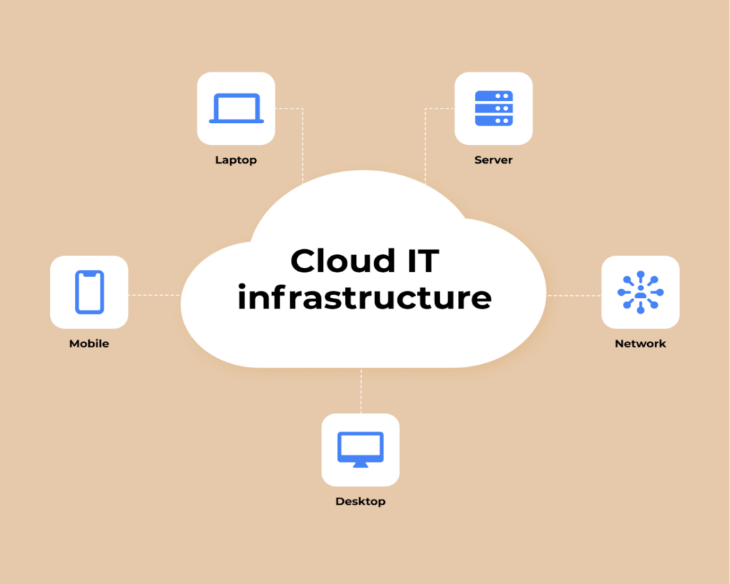


Рисунок 1.14. Облачная ИТ-инфраструктура (общий вариант)



Рисунок 1.15. Облачная структура

***Преимущества облачной ИТ-инфраструктуры***

1.      Отличная гибкость и масштабируемость

Гибкость и масштабируемость важны, если ваша компания стремится к быстрому росту. А облачные структуры не имеют ограничений по объему хранилища и гораздо большей мощности для вычислительных нужд.

Кроме того, гораздо проще масштабировать всю структуру вверх и вниз и при необходимости увеличивать или уменьшать количество серверов.

2.      Возможности автоматизации

Внедряя облачные технологии в свой бизнес, вы избавляетесь от головной боли, связанной с управлением оборудованием и вопросами безопасности.

Все эти операции передаются на аутсорсинг поставщику, который предоставляет вам облачную структуру, в то время как вы можете сосредоточиться на других важных аспектах бизнеса.

3.      Это экономически эффективно. Это в основном потому, что вы платите только за конкретную услугу, которую используете. Вам не нужно тратить деньги на оборудование, управление, обновления и другие аспекты, которыми сейчас занимается поставщик.

## **Собственная IT-инфраструктура и IaaS**

В зависимости от характера и размера бизнеса компании выбирают один из подходов.

**Собственная IT-инфраструктура** — это часто история крупных компаний, которым нужен полный контроль над оборудованием и данными. Нередко это банки и крупные государственные структуры. Для обеспечения собственного парка оборудования нужны значительные затраты: помимо закупки серверов, нужны люди, которые будут отвечать за их работу без сбоев. Позволить себе такое могут только крупные предприятия с доходом в несколько миллиардов рублей ежегодно.

Остальные компании — от микробизнеса до крупных организаций — все чаще **выбирают IaaS (инфраструктура как услуга)**. Сущность IaaS в том, что провайдер предоставляет клиенту готовые серверные мощности, а клиент может нагружать их и настраивать по своему плану.

IaaS исключает капитальные затраты на оборудование, регулярное обслуживание аппаратной части — все это заботы провайдера. Также провайдер обеспечивает сетевую связность серверов (сетевую инфраструктуру) и предоставляет готовые образы операционных систем для быстрого старта.

При этом клиент может влиять на конфигурацию сервера, его размещение, а также устанавливать на сервер любое необходимое ПО. IaaS — подходящая модель для компаний, если:

* хочется перейти от капитальных затрат к операционным и сэкономить,
* сложно обеспечить закупку собственного оборудования,
* нет квалифицированных специалистов для обеспечения работы инфраструктуры on-premises,
* есть плавающая («сезонная» нагрузка на серверы) — иногда нужно больше серверов, а иногда меньше,
* нужно обеспечить совместимость физических и облачных ресурсов (например, в случае распределенной инфраструктуры предприятия).

**3. Современные подходы к управлению производственными процессами**

функциональный подход

процессный подход (процессно-ориентированный, б**изнес-ориентированный подход)**

До начала 80-х годов подавляющее большинство компаний управлялись так называемым функциональным образом. *Суть* ***функционального подхода*** *к управлению заключается в управлении наборами функций.*

*Весь набор активностей организации разбивается на наборы функций (однородных и специализированных), которые сосредотачиваются в отдельных подразделениях.* Подразделения выстраиваются в иерархические структуры.

Управление такой структурой и сводится фактически к управлению функциями. Этот подход довольно прост и понятен сотрудникам организаций, легко усваивается и тиражируется по горизонтали и вертикали путем наращивания иерархий. Он был вполне оправдан в условиях постоянно увеличивающегося спроса, ускоряющего рост компаний, при наличии гарантированных рынков сбыта.

Однако в начале 80-х годов рынки наполнились, резко усилилась конкурентная борьба, борьба за клиента (покупателя). Организации стали активно искать пути радикального повышения эффективности деятельности. Новые условия продиктовали новые правила игры – снижение стоимости продукции при повышении ее качества и обеспечение гибкой и быстрой реакции компании на постоянно изменяющиеся внешние воздействия рынка.

Это привело к появлению новых подходов к управлению. Наибольшее развитие получил **процессный подход**. Суть подхода состоит в следующем. *Вся деятельность сотрудников компании состоит из двух видов активностей: повторяющихся (которые приходится осуществлять периодически либо в случае наступления определенных событий) и “разовых”, уникальных по составу, которые не повторяются в дальнейшем в таком виде.* Первый вид активностей называется ***процессами***, второй – ***мероприятиями***, проектами, программами и т.п. Если это так, то управлять деятельностью компании – значит управлять ее процессами (Process Management) и проектами (Project Management).

***В основе процессно-ориентированного подхода в управлении организацией лежит понятие процесса:*** это связанный набор повторяемых действий (функций), которые преобразуют исходный материал и (или) информацию в конечный продукт (услугу) в соответствии с предварительно установленными правилами.

**Бизнес-система** — связанное и полное множество [бизнес-процессов](https://piter-soft.ru/knowledge/glossary/process/biznes-protsess.html), реализуемых в рамках одной организационно оформленной [бизнес-единицы](https://piter-soft.ru/knowledge/glossary/process/biznes-edinitsa.html) при достижении её целей.

**Бизнес-процесс (БП)** включает в себя взаимосвязанные действия, которые реализуют одну (или несколько) из бизнес-целей компании в информационной системе компании.

**Бизнес-система** представляет собой категорию процессной модели предприятия, выраженную с помощью системного подхода в рамках процессного управления. Иногда бизнес-систему представляют в виде [дерева бизнес-процессов](https://piter-soft.ru/knowledge/glossary/process/derevo-biznes-protsessov.html), хотя на самом деле бизнес-система имеет более сложную структуру и обладает более сложными и не всегда однозначными связями, в отличие от простой иерархической структуры. Система бизнес-процессов пересекается с системой стратегического управления предприятием, а именно, с [деревом целей](https://piter-soft.ru/knowledge/glossary/process/derevo-tseley.html). Дерево целей разворачивается от миссии через стратегические цели к оперативным целям предприятия, а оперативные цели, в свою очередь, реализуются с помощью бизнес-процессов.

**Бизнес-система** проектируется с помощью модели предприятия, или системы [моделей бизнес-процессов](https://piter-soft.ru/knowledge/glossary/bpm/model-biznes-protsessa.html). Бизнес-процессы также могут быть связаны друг с другом посредством декомпозиции, т.е. они могут быть вложены друг в друга.

Существуют различные подходы к проектированию **бизнес-систем**. Среди них можно выделить следующие два:

* объектно-ориентированный, рассматривающий организацию как набор взаимосвязанных объектов, между которыми распределяется ответственность. К достоинствам этого подхода можно отнести устойчивость модели и четкое отражение организационной структуры
* функциональный, в соответствии с которым бизнес-система представляется набором функций, каждая из которых преобразует некоторые входные данные в выходные. Данный подход может использоваться в компаниях со слабо оформленной организационной структурой

**4. Современные подходы к построению архитектуры предприятия**

Появление в начале 80-х персональных компьютеров позволило автоматизировать ведение учета и обработку данных даже небольшим компаниям, не имеющим высококвалифицированного управленческого и технического персонала.

Сначала появились системы автоматизации внутренних (**бэк-офисных**) процессов, прежде всего производства (управление запасами и *автоматизация* управления производственными линиями) и бухгалтерского учета.

Затем пришел черед процессов взаимосвязи с внешней средой (процессы **фронт-офиса**): поставок, продаж, услуг, маркетинга.

В конце двадцатого века организации перешли к автоматизации перекрестных процессов, затрагивающих работу несколько подразделений, внедряя технологии управления взаимоотношениями с клиентами (*Customer* *Relationship* *Management* - *CRM*), и технологии управления цепочками поставок (*Supply* *Chain* *Management* - *SCM*). И, наконец, *вершина* *пирамиды* - это корпоративное управление.

Основой КИС предприятий на современном этапе являются так называемые *системы планирования ресурсов предприятий* (*Enterprise* Recourse *Planning* - *ERP*). Мировой *опыт* свидетельствует, что умело выбранная и внедренная *ERP*-система существенно улучшает управляемость предприятием и повышает эффективность его работы. Построение *корпоративной информационной системы* должно начинаться с анализа структуры управления организацией и соответствующих потоков данных и информации. *Координация* работы всех подразделений организации осуществляется через органы управления разного уровня.

Под **управлением** понимают обеспечение поставленной цели при условии реализации следующих основных функций: организационной, плановой, учетной, анализа, контрольной, стимулирования.

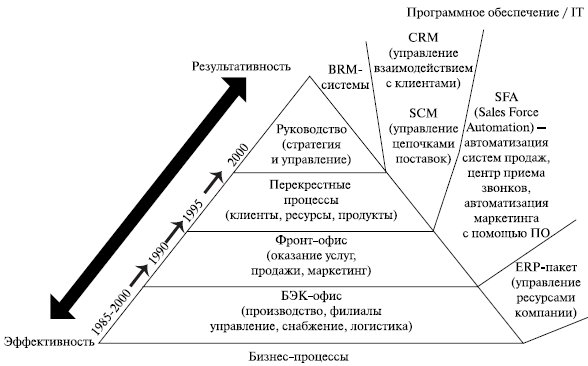


Рисунок 1.16. Эволюция развития систем управления предприятия

В настоящее время в мировой практике для обозначения полнофункциональных интегрированных *АСУ*, используемых предприятиями, применяют следующие названия:

*MRP* (Material Requirement *Planning* - Планирование материальных потребностей),

*MRP II* (*Manufacturing* *Resource* *Planning* - Планирование производственных ресурсов),

*ERP*-система (*Enterprise* *Resource* *Planning* -Планирование ресурсов предприятия),

*ERP*-II и *CSRP* (*Customer* *Synchronized* *Relationship* *Planning* - Планирование ресурсов, синхронизированное с покупателем).

Какая-либо однозначная и общепринятая общая классификация ИТ- предприятий отсутствует. Возможный вариант обобщенной структуры современных информационных технологий, внедряемых на промышленных производствах различного типа, приведен на [рис.](https://intuit.ru/studies/courses/1055/271/lecture/6882?page=1#image.9.2) 2, на котором сделаны следующие общепринятые сокращения:

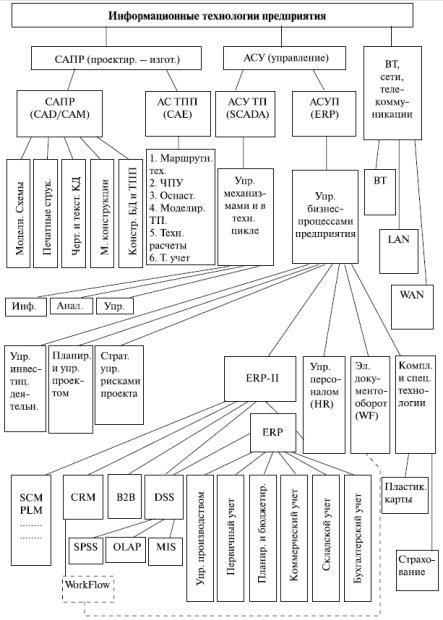


Рисунок 1.17.Обобщенная структура информационных технологий предприятия

* *САПР* - системы автоматизированного проектирования/изготовления (Computer Aided Design / Computer Aided *Manufacturing* - *CAD*/*CAM*);
* АСТПП - *автоматизированные системы технологической подготовки производства* (Computer Aided Engineering - *CAE*);
* *АСУТП* - *автоматизированные системы управления технологическими процессами* (Supervisory Control And *Data Acquisition* - *SCADA*);
* АСУП - комплексная *автоматизированная система управления* предприятием (Enterprise Resource *Planning* - *ERP*)
* WF - *потоки работ* (*WorkFlow*);
* *CRM* - управление отношениями с клиентами;
* *B2B* - электронная торговая площадка ("онлайновый бизнес");
* *DSS* - поддержка принятия управленческих решений;
* SPSS - *статистический анализ* данных;
* *OLAP* - анализ многомерных данных;
* *MIS* - управляющая информационная система, (*АРМ*) руководителя;
* *SCM* - управление цепями поставок;
* *PLM* - управление жизненным циклом продукции (характерно для дискретного производства);
* *ERP*-II - расширение *ERP*-системы за контуры производства (т. е. *ERP* + *CRM* + *B2B* + *DSS* + *SCM*+ *PLM* и т. п.);
* HR - "*Управление персоналом*"; можно рассматривать и как самостоятельную задачу, и как входящую в состав *ERP* (что и отображено на рисунке в виде двух связей);
* *LAN* - *локальные вычислительные сети* (Local Area Net);
* *WAN* - глобальные (внешние) сети и *телекоммуникации* (Wide Area Net).

**2.1. MRP (Materials Resource Planning - планирование материальных ресурсов)**

К концу 80-х годов идея создания единой модели данных в рамках целого предприятия заинтересовала ряд международных промышленных компаний, которые искали способ упростить управление производственными процессами. Первым шагом в данном направлении стала разработка **концепции MRP** (Materials Resource Planning - планирование материальных ресурсов), рассматривавшей ***планирование материалов для производства***.

Основная цель концепции MRP заключалась в минимизации издержек, связанных со складскими запасами (в том числе и на различных участках производства). В основе этой концепции лежит понятие ВОМ (Bill Of Material - спецификация изделия, ответственность за которую возложена на конструкторский отдел), отражающее зависимость спроса на сырье, полуфабрикаты и другие продукты от плана выпуска готовой продукции. При этом очень важную роль играет время, для учета которого необходимо иметь четкое представление о технологической цепочке выпуска продукции, то есть знать, какова последовательность и длительность операций. На основании плана выпуска продукции, ВОМ и технологической цепочки осуществляется расчет потребности в материалах к конкретным срокам.

**2.2. MRPII (Manufacturing Resource Planning - планирование производственных ресурсов)**

Однако у концепции MRP есть серьезный недостаток. Дело в том, что при расчете в рамках этой концепции потребности в материалах не учитываются ни имеющиеся производственные мощности, ни их загрузка, ни стоимость рабочей силы. Этот недостаток был исправлен в **концепции MRPII** (Manufacturing Resource Planning - планирование производственных ресурсов). ***MRPII позволяла учитывать и планировать все производственные ресурсы предприятия*** - сырье, материалы, оборудование, персонал и т.д.

По мере развития концепции MRPII к ней постепенно добавлялись возможности учета остальных затрат предприятия. Так появилась концепция **ERP**

**2.3.** **ERP** (**Enterprise Resource Planning** - **планирование ресурсов предприятия)**

Концепция **ERP** (**Enterprise Resource Planning** - **планирование ресурсов предприятия**), называемая иногда также планированием ресурсов в масштабе предприятия (Enterprise-wide Resource Planning). В основе ERP *лежит принцип создания единого хранилища данных (репозитария), содержащего всю деловую информацию, накопленную организацией в процессе ведения бизнеса, в частности, финансовую информацию, данные, связанные с производством, управлением персоналом*, и любые другие данные. Наличие репозитария избавляет от необходимости передавать данные от приложения к приложению.

Стандарт ERP позволил объединить все ресурсы предприятия и повысить эффективность управления ими.



Рисунок 1.18. Схема взаимодействия процессов

В настоящее время практически все современные западные системы управления производством базируются на концепции ERP и отвечают ее рекомендациям.

На мировом рынке сейчас предлагается свыше 500 систем класса MRPII - ERP. Развитие этого рынка идет очень быстрыми темпами - число внедрений таких систем в мире растет на 35-40% в год. На отечественном же рынке сейчас присутствуют около десятка западных систем и три-четыре отечественные системы класса КИС (корпоративные информационные системы). Они представлены в таблице 1.

Тиражируемые интегрированные системы управления предприятием (ИСУП),   
представленные на рынке  СНГ.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Название тиражируемой ИСУП** | **Класс** | **Фирма-поставщик в России** |
| ИСУП для крупных предприятий | R/3 | ERP | **SAP** |
| Baan | ERP | Альфа-Интегратор - Баан Евразия |
| Oracle Applications | ERP | Oracle CIS |
| OneWorld J.D. Edwards | ERP | Robertson & Blums |
| ИСУП для средних предприятий | SyteLine (разработчик — Symix) | CSRP | Socap |
| MAX (разработчик - MAX International) | ERP | ICL-КПО ВС (Казань) |
| Mfg/Pro (разработчик - QAD) | ERP | BMS |
| iRenaissance CS (разработчик - Ross Systems) | ERP | «Интерфейс» |
| Axapta (разработчик - Damgaard, Дания) | ERP | Columbus IT Partner |
| ИСУП для малых и средних предприятий | Concorde XAL (разработчик - Damgaard, Дания) | ERP | Columbus IT Partner |
| Exact | ERP | Exact Software |
| Platinum ERA | ERP | Platinum Software |
| Scala | ERP | Scala CIS |
| SunSystems (фирмы Systems Union) + RB Manufacturing (разработчик – Robertson & Blums) | MRP | Robertson & Blums |
| «Галактика» |  | «Галактика» |
| М-2 | MRP | «Клиент-серверные технологии» |
| АС+ | MRP | «Борлас» |
| «1С:Предприятие» (с модулем «Производство») |  | **«1С»** |

**2.4. CSRP (Customer Synchronized Resource Planning)**

Стандарт системы управления предприятиями - **CSRP** (Customer Synchronized Resource Planning) - помимо всего прочего охватывает и взаимодействие с клиентами, оформление нарядов/заказов и технических заданий, поддержку заказчика на местах и т.д. Таким образом, если стандарты MRP, MRPII и ERP ориентированы на внутреннюю организацию предприятия, то **стандарт CSRP включает в себя полный цикл - от проектирования будущего изделия, с учетом требований заказчика, до гарантийного и сервисного обслуживания после продажи**. Суть концепции CSRP главным образом состоит в том, чтобы интегрировать заказчика (клиента, покупателя) в систему управления предприятием. Согласно *данной концепции не отдел сбыта, а непосредственно сам покупатель размещает заказ на изготовление продукции, сам отвечает за правильность его исполнения и при необходимости отслеживает соблюдение сроков производства и поставки*.

При этом само предприятие может очень четко отслеживать тенденции спроса на его продукцию.

**2.5. Системы управления активами и фондами (ЕАМ)**

**Системы управления активами и фондами EAM** (Enterprise Asset Management) представляет собой интегрированный, ориентированный на активы/фонды подход к управлению производственной деятельностью, позволяющий предприятиям с большими вложениями в основные средства значительно увеличить чистую прибыль.

Системы управления активами и фондами предприятия — это новая и быстро растущая категория систем, отвечающая специфическим требованиям к технологиям капиталоемких предприятий. В табл. 2 приведены основные отличия подхода ЕАМ от MRP/ERP.

*Под активами и фондами* понимается все, что требуется для обеспечения работы предприятия (люди, деньги, оборудование, машины, материально-производственные запасы, здания и сооружения, инструменты и т. д.).

ЕАМ-системы применяются в таких отраслях, как горнодобывающая промышленность, обрабатывающая промышленность, транспорт и перевозки, оборонная промышленность, а также на предприятиях по эксплуатации инженерных коммуникаций, коммунальных предприятиях, вырабатывающих, поставляющих и распределяющих газ, воду, электроэнергию и т.п. Они обеспечивают управление полным жизненным циклом фондов/активов с использованием управления прогнозированным и превентивным техническим обслуживанием, управления материалами, кадрами, производственной и финансовой деятельностью. Обычно такие системы включают следующие основные модули:

* управление производственной деятельностью, техническим обслуживанием и ремонтом;
* управление материалами;
* управление персоналом;
* управление финансами.

*Таблица 2.*  **Основные отличия подхода ЕАМ от MRP/ERP**

|  |  |
| --- | --- |
| **MRP/ERP** | **ЕАМ** |
| Ориентация на продукт | Ориентация на процесс |
| Производство оборудования | Эксплуатация оборудования |
| Материалы, необходимые для производства новых товаров | Материалы, требуемые для поддержания существующих активов |
| Рабочие задания инициируются необходимостью создать новые изделия | Рабочие задания инициируются необходимостью в ремонте, перестройке или техническом обслуживании активов |
| Статичный лист заказа материалов маленькие изменения в типах материалов в зависимости от задания | Динамичный список заказа, основанный на различных типах проектов, работ и конфигурации оборудования |
| Большие объемы относительно небольшого количества деталей от нескольких производителей | Небольшие объемы очень большого числа деталей от многих производителей и поставщиков |
| В производстве людей занято больше, чем в техническом обслуживании | В техническом обслуживании людей занято больше, чем в производстве |
| Материалы закупаются на основе прогноза продаж | Материалы закупаются на основе спланированного технического обслуживания, ремонтной истории и статистики использования материально  производственных запасов |
| Относительно простой список материалов, определяемый заказами на отделку продукции | Крайне сложный список материалов для разных типов оборудования и работ |
| Легко предсказать потребность в материалах на основе прогноза продаж | Трудно предсказать потребность в материалах на основе выходов из строя и состояния оборудования |

**2.6. CRM системы (Customer Relationship Management) - Системы управления продажами, маркетингом, отношениями с клиентами.**

**CRM системы** (Customer Relationship Management) - Системы управления продажами, маркетингом, отношениями с клиентами. Управление отношениями с клиентами (Customer Relations Management, CRM) — это стратегия, ***основанная на применении новых управленческих и информационных технологий, с помощью которых компании аккумулируют знания о клиентах для выстраивания взаимовыгодных отношений с ними***. Подобные отношения способствуют увеличению прибыли, т.к. привлекают новых клиентов и помогают удержать старых.

**Информационная система CRM.** Стратегия CRM *реализуется с помощью специального набора программного обеспечения (ПО) и технологий, позволяющих автоматизировать, а значит, совершенствовать бизнес-процессы в сфере продаж, маркетинга и обслуживания клиентов*. Это дает возможность компании обращаться к заказчикам услуг с предложениями в наиболее удобный момент времени и по наиболее удобным каналам связи. Система CRM облегчает координацию действий различных отделов, обеспечивая их общей платформой для взаимодействия с клиентами, и дает каждому из них доступ к полной информации о них, что способствует наилучшему удовлетворению потребностей клиентов.

Основная функциональность большинства CRM-систем оформляется в виде следующих самостоятельных компонентов:

- автоматизация продаж (Sales Force Automation — SFA);

- автоматизация маркетинга (Marketing Automation — MA);

- автоматизация обслуживания клиентов (Customer Service & Support — CSS);

- средства анализа и построения отчетов.

Основой системы **CRM** являются приложения ***автоматизации продаж*** (Sales Force Automation, SFA). На них возлагаются следующие функции:

* ведение календаря событий и планирование работы;
* управление контактами (благодаря ему ни один важный звонок или личное обращение не будут пропущены);
* работа с клиентами (каждый клиент будет обслужен на высочайшем уровне, благодаря зафиксированной истории взаимодействия с ним);
* мониторинг потенциальных продаж (ни одна потенциальная возможность не будет упущена, каким бы плотным не было расписание сотрудника);
* поточная организация продаж (эффективное управление циклом продаж);
* повышение точности прогнозов продаж;
* автоматическая подготовка коммерческих предложений (освобождает сотрудников от рутинной работы);
* предоставление информации о ценах;
* автоматическое обновление данных о размере бонуса в зависимости от выполнения поставленных задач;
* предоставление актуальной информации о состоянии дел в региональных представительствах;
* формирование отчетов (эффективный инструментарий автоматического создания отчетов по результатам деятельности);
* организация продаж по телефону (создание и распределение списка потенциальных клиентов, автоматический набор номера, регистрация звонков, прием заказов).

SFA дополняется sales-конфигуратором, позволяющим конфигурировать те или иные продукты из компонентов. Правила конфигурирования заложены в самом приложении, что дает возможность клиентам производить покупки через Интернет.

***Автоматизация маркетинга.*** В современных CRM-системах SFA-приложения дополняются **средствами автоматизации маркетинга** (Marketing Automation, **MA**), они позволяют:

* организовывать маркетинговые кампании (предусмотрены инструменты планирования, разработки, проведения и анализа результатов маркетинговых акций, как традиционных, так и через Интернет);
* создавать маркетинговые материалы и управлять ими (в том числе заниматься автоматической рассылкой);
* генерировать список целевой аудитории (создание списков потенциальных клиентов и их распределение между торговыми представителями);
* отслеживать бюджетирование и прогнозирование результатов маркетинговых кампаний;
* вести маркетинговую энциклопедию (репозиторий информации о продуктах, ценах и конкурентах).

***Автоматизация обслуживания клиентов.*** Приложения автоматизации обслуживания и поддержки клиентов (Customer Service & Support, **CSS**) в последнее время приобрели первостепенное значение, так как в условиях жесткой конкуренции удержать прибыльного клиента можно прежде всего благодаря высокому качеству обслуживания. Как правило, к этой категории приложений относятся ***средства обработки вызовов и самообслуживания через Интернет***. Приложения CSS позволяют удовлетворять индивидуальные потребности заказчиков быстро, точно и эффективно, обеспечивая выполнение следующих функций:

* мониторинг потребностей клиента (сотрудники отдела обслуживания всегда в курсе проблем и предпочтений того или иного покупателя услуг);
* мониторинг прохождения заявок (процесс отслеживается автоматически);
* мониторинг мобильных продаж (в любой момент времени можно получить информацию о качестве выполнения услуги, ее стоимости, удовлетворенности клиентов, сроках выполнения заявки и др.);
* ведение базы знаний (эффективный инструмент снижения себестоимости услуг — большинство проблем могут быть решены во время первого звонка клиента);
* контроль за исполнением сервисных соглашений (автоматическое отслеживание сроков и условий);
* управление запросами клиентов с помощью присвоения приоритетов.

Приложения CSS превращают отделы обслуживания клиентов из затратных в прибыльные. Будучи интегрированными с приложениями SFA и МА, они способствуют тому, чтобы каждый контакт клиента с компанией был использован для продажи дополнительных услуг (cross-sell) и более дорогих продуктов (up-sell).

**Современные классификации выделяют три следующих типа CRM-систем:**

* оперативная CRM — обеспечивает оперативный доступ к информации по конкретному клиенту в процессе взаимодействия с ним в рамках традиционных бизнес-процессов продажи и послепродажного обслуживания. Ее функциональность охватывает маркетинг, продажи и сервис, что соответствует стадиям привлечения клиента, акта совершения сделки и послепродажного обслуживания, т. е. все те точки контакта, где осуществляется взаимодействие предприятия с клиентом;
* аналитическая CRM — обеспечивает синхронизацию разрозненных массивов данных и поиск статистических закономерностей в этих данных для выработки наиболее эффективной стратегии маркетинга, продаж, обслуживания клиентов и т. п.
* коллаборационная CRM — предоставляет клиенту возможность непосредственного участия в процессе дизайна, производства доставки и обслуживания продукта (сбор предложений клиентов при дизайне продукта, доступ клиентов к прототипам продукции и возможность обратной связи, реверсивное ценообразование — когда клиент описывает требования к продукту и определяет цену, которую он готов заплатить, а производитель реагирует на эти предложения).

В настоящее время прослеживается **тенденция включения функциональности CRM-компонентов в ERP-системы различного класса** (от систем, предназначенных для крупных предприятий, до систем, ориентированных на средние и мелкие). Подобная интеграция позволяет:

* обеспечить совместное использование данных о клиенте, собираемых разными компонентами;
* расширить рамки системы управления ресурсами предприятия, включив в нее заказчиков, поставщиков, партнеров, и, как следствие, снизить издержки предприятия на продажи, поставки, маркетинг;
* ориентировать бизнес-процессы предприятия на наиболее полное и оперативное удовлетворение потребностей клиента за счет включения его заказов в систему производственного планирования;
* улучшить каналы взаимодействия клиента с предприятием, сделав их максимально удобными и предоставив клиенту возможности персонального обслуживания и самообслуживания.

**2.7 Системы управления цепочками поставок (SCM)**

Системы управления цепочками поставок SCM (Supply Chain Management) поддерживают технологию управления, реализующую концепцию CSRP (Customer Synchronized Resource Planning), которая предполагает наличие *возможностей управления внешними по отношению к предприятию элементами производственной цепочки, а именно системой материальных потоков в сети поставщиков сырья и комплектующих*. В ведении SCM-систем находится рассмотрение *логистических операций* на протяжении полного жизненного цикла изделия, т.е. процесс разработки, производства, продажи и послепродажного обслуживания.

Следует отметить, что **управление цепочками поставок является одним из наиболее сложных и трудоемких бизнес-процессов в условиях многопрофильных предприятий. Интеграция поставщиков, производителей продукции, дилеров, транспортных и финансовых компаний, участвующих в цепочках поставок**, оказывается крайне затруднительной из-за несовместимых организационных, управленческих, информационных и других систем.

Идея управления цепочками поставок достаточно проста, в ее основе лежат следующие очевидные факты:

* стоимость товара формируется на протяжении всей цепочки поставок;
* на стоимость товара оказывает влияние не только и не столько эффективность операций по конкретной продаже, сколько общая эффективность операций по всей цепочке поставок;
* наиболее управляемыми с точки зрения стоимости являются начальные звенья цепочки поставок, связанные с производством товара, а наиболее чувствительными — заключительные звенья, связанные с его продажей.

Поэтому в SCM-системах акцент делается на поддержке процессов планирования производства и дистрибуции (проектирование сети цепочек, планирование и прогноз спроса, планирование снабжения и сбыта, планирование и составление графиков производства), а также поддержке процессов обеспечения выполнения поставок с ориентацией на ежедневное управление сбытом (ресурсы, перевозки, логистика, склад).

Традиционная функциональность SCM-системы включает в себя:

- планирование и прогнозирование спроса;

- выбор поставщиков и управление закупками;

- обработку/выполнение заказа и послепродажное обслуживание;

- управление складами;

- управление отгрузкой и транспортировкой;

- производственную логистику;

- расчеты;

- анализ эффективности отдельных элементов системы поставок.

Внедрение SCM-систем позволяет предприятию сократить стоимость и время обработки заказа, время выхода товара на рынок, затраты на закупки сырья и комплектующих, уменьшить складские запасы, сократить производственные затраты и в конечном счете увеличить прибыль.

**2.8. Системы управления бизнес-правилами, BRM-системы (Business Rule Management System)**

Стратегический уровень руководства предприятием в иерархии информационных систем использует в числе прочих **системы управления бизнес-правилами, BRM-системы (Business Rule Management System (BRMS)), предназначенные для создания и использования модели, формально описывающей логику принятия решений в виде системы бизнес-правил – утверждений в терминах бизнес-пользователей, указывающих на выполнение некоторых действий в случае выполнения определенных условий**.

**BRMS** ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *Business Rule Management System* — система управления бизнес-правилами) — [информационная система](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0), используемая для ведения, поддержки и исполнения бизнес-правил компании.

В традиционных ИТ-системах бизнес-правила заложены внутрь технического кода, который доступен для понимания только ИТ-специалистам, что затрудняет их понимание бизнес-пользователями и требует перенастройки всего ИТ-цикла даже при небольших изменениях.

В общем виде бизнес-правило представляется в форме утверждения вида:

*Если (условия), то (список действий), иначе (альтернативный список действий).*

В системах управления бизнес-правилами предприятия они рассматриваются как управляемые на протяжении всего их жизненного цикла активы, что предполагает использование кардинально отличающихся от традиционного подхода стратегий, основными из которых являются:

1) выделение бизнес-правил в централизованное хранилище правил;

2) возможность для бизнес-пользователей быть авторами изменений.

Бизнес-правила в BRMS объединяются в блоки в виде таблиц или деревьев решений (графическое изображение процесса принятия решений, в котором отражены альтернативные решения, альтернативные состояния среды, соответствующие вероятности и выигрыши для любых комбинаций альтернатив и состояний среды (рисунок 1.19)).

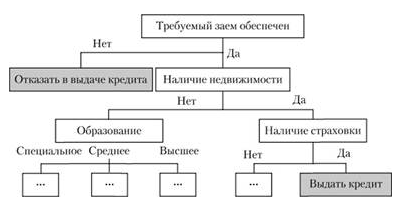


Рисунок 1.19 Пример дерева решений на выдачу кредита

BRM-система позволяет гибко автоматизировать логику принятия решений в рамках бизнес-процессов компании на основе совершенствования уже работающих в организации приложений управления бизнес-правилами и эффективной совместной деятельности бизнес-пользователей и ИТ-специалистов. Доступность правил позволяет бизнес-пользователям быстро выполнять изменения. Результатом становится повышение гибкости бизнеса, быстрое реагирование на динамику рынка и предоставление клиентам новых возможностей. BRMS хранит всю историю изменения бизнес-логики.

Основными преимуществами использования BRMS являются:

* повышение адаптивности и оперативности реакции информационной среды компании на изменения в стратегии бизнеса за счет вовлечения бизнес-пользователей в процесс управления бизнес-правилами;
* снижение затрат на адаптацию систем к изменившимся условиям бизнеса;
* повышение прозрачности логики принятия решений в компании и ее аудита;
* возможность реализации системного подхода к управлению принятием решений в организации.

Использование систем управления бизнес-правилами является важной частью системы управления принятием решений в организации (СППР). Примеры систем BRM: ILOG JRules, JBoss Drools.

**Критерии выбора систем управления ресурсами предприятий**

В настоящее время на рынке представлено большое количество систем управления ресурсами предприятий (отличающихся стоимостью, количеством поддерживаемых функций, специализацией по отраслям и / или видам производства, средствами анализа и формирования отчетности и т. п.), характеризующихся управлением практически всеми видами деятельности и всеми видами ресурсов, первичностью производственных процессов (а не бухгалтерского учета), возможностью охвата корпораций, холдингов и т. п., а не отдельных предприятий. Перечислим наиболее значимые критерии выбора такой системы:

- соответствие требованиям вышеперечисленных общепринятых стандартов и методологий управления производством;

- наличие отраслевой специфики;

- удовлетворение корпоративным стандартам (стандартам уровня предприятия);

- наличие высокоуровневых средств разработки, позволяющих добавлять новую функциональность и / или модифицировать существующую;

- возможность интеграции с другими приложениями (в том числе и с унаследованными системами);

- наличие реализованных проектов для аналогичных предприятий;

- практический опыт реализации проектов компанией, осуществляющей внедрение.

***Основная литература***

1. Архипенков С. Лекции по управлению программными проектами// Электронный ресурс:http://citforum.ru/SE/project/

2. Основы управления ИТ-проектами [Электронный ресурс]: учебное пособие / сост. Е.Р. Кирколуп, Ю.Г., Скурыдин, Е.М. Скурыдина. – Барнаул : АлтГПУ, 2017// Электронный ресурс:http://library.altspu.ru/dc/pdf/kirkolup.pdf

3. Рыбалова Е.А. Управление проектами: учебное пособие / Е.А. Рыбалова. - Томск: ФДО, ТУСУР, 2015. - 205 с.

4. Бухарев Н.Р. Введение в гибкое управление проектами. Краткий конспект лекций// tulpar.kfu.ru/course/view.php?id=2284

[5. Коптелов А., Никитин И., Цулая М. Гибкая процессная методология Agile](https://www.intuit.ru/studies/courses/3590/832/info). - М.: ИНТУИТ // Электронный ресурс: // www.intuit.ru/studies/courses/3590/832/print\_lecture/31170

6. Топ-7 методов управления проектами: Agile, Scrum, Kanban, PRINCE2 и другие// www.pmservices.ru/project-management-news/top-7-metodov-upravleniya-proektami-agile-scrum-kanban-prince2-i-drugie/

### 7.[Грекул](https://www.intuit.ru/intuituser/userpage/v_i_grekul) В., [Коровкина](https://www.intuit.ru/intuituser/userpage/n_l_korovkina) Н., [Куприянов](https://www.intuit.ru/intuituser/userpage/y_v_kupriyanov) Ю. [Методические основы управления ИТ-проектами](https://www.intuit.ru/studies/courses/646/502/info). - М.: ИНТУИТ // Электронный ресурс: https://www.intuit.ru/studies/courses/646/502/info

### ***Дополнительная литература:***

8. Сериков А. Основы управления проектами. - М.: ИНТУИТ. Электронный ресурс: www.intuit.ru/studies/professional\_retraining/965/courses/272/info

9. Управление проектами: Учебное пособие / М.В. Романова. - М.: ИД ФОРУМ: НИЦ ИНФРА-М, 2014

10. Руководство к Своду знаний по управлению проектами РМВОК (A Guide to the Project Management Body of Knowledge PMBOK®) — 4-е изд. — Project Management Institute, Inc., 2008. — 464 с.

11. https://products.office.com/

**Лекция 2. Программная инженерия. Жизненный цикл программного продукта**

*План:*

1. Введение в программную инженерию

2 Процесс разработки программного обеспечения.

3. Стандартизация программной инженерии

3.1 Виды стандартов

3.2 Руководства по разработке ПО

* PMBoK
* SWEBOK

4 Модели жизненного цикла программных продуктов

* + каскадная (ступенчатая, водопадная);
  + спиральная
  + эволюционная (итеративная, инкрементальная, RAD-модель)

5 Методологии процесса разработки ПО

5.1. Методологии Последовательного подхода

#### ГОСТы

#### SW-CMM

#### RUP

#### MSF

#### PSP/TSP

5.2 Методологии подхода гибкой разработки Agile

* Экстремальное программирование (XP)
* Методология [Kanban](https://www.atlassian.com/ru/agile/kanban)
* Методология [Scrum](https://www.atlassian.com/ru/agile/scrum)

5.3 Сравнение классических и гибкого подходов

**1.** **Введение в программную инженерию**

Термин **– software engineering (программная инженерия)** — впервые был озвучен в октябре 1968 года на конференции подкомитета НАТО по науке и технике (г.Гармиш, Германия), которая была посвящена этим вопросам.

Связана она была с тем, что тогда окрестили как *«программный кризис» — программирование не успевало за быстрым ростом технических средств* (дала сбои система резервирования и управления потоками в авиатранспорте).

*Основные причины возникновения кризиса заключались в следующем:*

• реальные программные проекты выполнялись с отставанием от графика или с превышением сметы расходов;

• программные продукты не обладали требуемыми функциональными возможностями, производительность программного продукта низка, качество не удовлетворяет потребителей;

• возникающая в таких условиях конкурентная борьба существенно затрудняла возможность выпуска качественного программного обеспечения в приемлемые сроки;

• методы и процессы программирования, которые эффективно работали для одного человека или для небольшой команды при разработке программ умеренных размеров, не приводили к успеху при разработке крупных и сложных систем.

**Появление неудачных программных проектов объяснялось следующими факторами:**

• нечеткая и неполная формулировка требований к ПП,

• недостаточное вовлечение пользователей в работу над проектом,

• отсутствие необходимых ресурсов,

• неудовлетворительное планирование,

• частое изменение требований и спецификации, новизна используемой технологии проектирования ПП,

• отсутствие грамотного управления проектом, недостаточная поддержка со стороны высшего руководства.

Постепенно общество пришло к пониманию, что отношение к разработке программного обеспечения должно быть более серьезным, возникла объективная потребность в соответствующей регламентации и стандартизации этих процессов.

Сегодня программная инженерия является прикладной отраслью фундаментальной математики и информатики, экономики, менеджмента.

Как инженерная дисциплина, программная инженерия охватывает все аспекты создания ПО, начиная от формирования требований до создания, сопровождения и снятия с эксплуатации ПО, а также включает инженерные методы оценки трудозатрат, стоимости, производительности и качества ПО.

Определения различны:

**Программная инженерия** — это область компьютерной науки и технологии, которая занимается построением программных систем…

**Программная инженерия** — инженерная дисциплина, которая связана со всеми аспектами производства программного обеспечения (ПО) от начальных стадий создания спецификации до поддержки системы после сдачи в эксплуатацию.

**Программная инженерия** — методология разработки, внедрения и сопровождения в рамках имеющегося бюджета программного обеспечения с заданными уровнем качества и сроками реализации.

**Каждое из этих определений содержит отдельные аспекты, повлиявшие на общее понимание программной инженерии. Однако им всем присуща одна общая черта — *программная инженерия это нечто большее, чем просто написание программного кода*.**

При создании новых продуктов существует **набор требований**, являющихся не только общими к любой инженерной деятельности, но и существенными для описания основ инженерии как таковой.

**Основными из этих требований являются**: ***удовлетворение заданным (возможно, неформальным) функциональным требованиям, накладываемым пользователями; явным и неявным требованиям по эксплуатационным параметрам и ресурсопотреблению; явным и неявным критериям дизайна продукта; требованиям к самому процессу разработки (продолжительности, стоимости, качеству).***

В сферу программной инженерии попадают все вопросы и темы, связанные с организацией и улучшением процесса разработки *ПО*, управлением коллективом разработчиков, разработкой и внедрением программных средств поддержки жизненного *цикла* разработки *ПО*. 

*Программная инженерия* использует достижения информатики, тесно связана с системотехникой, часто предваряется бизнес-реинжинирингом. Немного подробнее об этом контексте программной инженерии.

**Информатика** (*computer* *science*) – это свод теоретических наук, основанных на математике и посвященных формальным основам вычислимости. Сюда относят математическую логику, теорию грамматик, методы построения компиляторов, математические формальные методы, используемые в верификации и модельном тестировании и т.д. Трудно строго отделить программную инженерию от информатики, но в целом направленность этих дисциплин различна. *Программная инженерия* нацелена на решение проблем производства, *информатика*  – на разработку формальных, математизированных подходов к программированию.

**Системотехника** (*system engineering*) объединяет различные инженерные дисциплины *по* разработке всевозможных искусственных систем – энергоустановок, телекоммуникационных систем, встроенных систем реального времени и т.д. Очень часто *ПО* оказывается частью таких систем, выполняя задачу управления соответствующего оборудования. Такие системы называются *программно-аппаратными*, и участвуя в их создании, программисты вынуждены глубоко разбираться в особенностях соответствующей аппаратуры.

**Бизнес-реинжиниринг** (business reengineering) – в широком смысле обозначает модернизацию бизнеса в определенной компании, внедрение новых практик, поддерживаемых соответствующими новыми информационными системами. При этом акцент может быть как на внутреннем переустройстве компании так и на разработке нового клиентского сервиса (как правило, эти вопросы взаимосвязаны). Бизнес-реинжиниринг часто предваряет разработку и внедрение информационных систем на предприятии, так как требуется сначала навести определенный порядок в делопроизводстве, а лишь потом закрепить его информационной системой. – это Бизнес-ориентированный подход к ИТ-инфраструктуре предприятия.

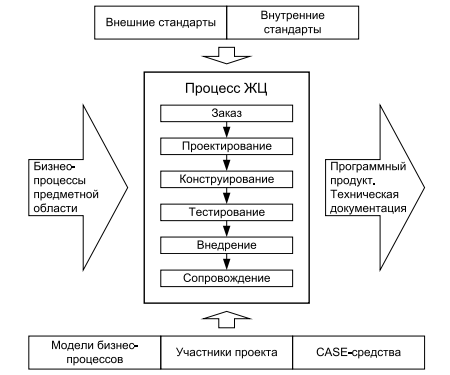
*Связь* программной инженерии (как области практической деятельности) с информатикой, системотехникой и бизнес-реинжинирингом показана на [Рис. 1](https://intuit.ru/studies/courses/497/353/lecture/8405?page=1#image.1.1).



Рисунок 2.1. Связь программной инженерии с информатикой, системотехникой и бизнес-реинжинирингом

**2 Процесс разработки программного обеспечения.**

**Процесс разработки программного обеспечения представляет собой специфический технологический процесс преобразования исходных требований заказчика о предметной области в готовый программный продукт и состоит из совокупности взаимосвязанных этапов (технологических операций)** (рис. 2.2).



*Рисунок 2.2. Модель технологического процесса создания программного продукта*

В процессе преобразования участвуют специалисты, выполняющие определенную работу с помощью специфических инструментальных средств и предметов производства.

**На вход** этого процесса в виде требований поступают сведения о предметной области.

**На выходе** процесса — разработанное программное обеспечение и техническая документация.

**Механизмами, обеспечивающими выполнение процесса**, являются специалисты и используемые ими инструментальные средства.

Суть **методологии программной инженерии** состоит в применении систематизированного, научного и предсказуемого процесса проектирования, разработки и сопровождения программных средств.

**Метод программной инженерии** — это структурный подход к созданию ПО, который способствует производству высококачественного продукта эффективным в экономическом аспекте способом.

В этом определении есть две основные составляющие:

(а) создание высококачественного продукта и

(б) экономически эффективным способом.

Иными словами, **метод** – это то, что обеспечивает решение основной задачи программной инженерии: создание качественного продукта при заданных ресурсах времени, бюджета, оборудования, людей.

Начиная с 1970-х годов создано достаточно много **методов разработки ПО.**

Наиболее известны:

· Метод структурного анализа и проектирования Том ДеМарко (1978),

· Метод сущность-связь проектирования информационных систем Чен (1976)

· Метод объектно-ориентированного анализа Буч (1994), Рамбо (1991).

***Метод программной индустрии основан на идее создания моделей ПО с поэтапным преобразованием этих моделей в программу – окончательную модель решаемой задачи.*** Так, на этапе спецификаций создается модель – описание требований, которая далее преобразуется в модель проекта ПО, проект – в программный код.

При этом важно, чтобы модели метода представлялись графически с помощью некоторого языка представления моделей.

Нет идеальных методов, все они применимы только для тех или иных случаев.

**CASE-инструменты**

CASE - Computer Aided System Engineering - различного рода инструментальные программы, используемые для поддержки процесса создания программ.

*Современные CASE-технологии представляют собой методологию создания программных продуктов, а также набор инструментальных средств моделирования, проектирования и разработки, позволяющих в наглядной форме моделировать предметную область, анализировать эту модель на всех этапах создания и сопровождения ПП, разрабатывать приложения в соответствии с потребностями пользователей.*

В основу большинства CASE-технологий положены методологии структурного или объектно-ориентированного анализа и проектирования.

CASE средства могут быть классифицированы по нескольким признакам:

1) По уровню применения:

* Upper CASE -средства анализа требований
* Middle CASE - средства проектирования
* Low CASE - средства разработки приложений

2) Специализированные:

* Средства проектирования баз данных
* Средства реинжиниринга (восстановления) модели (формирование ERD на основе анализа схем БД или формирования диаграмм на основе анализа программных кодов)

3) Вспомогательные

* *Планирования и управления проектом*
* Конфигурационного управления
* Тестирования

4) Интегрированные CASE охватывают все этапы и процессы создания ПО от анализа требований до тестирования и выпуска документации. Интегрированные CASE выступают, как правило, в виде набора согласованных по интерфейсу средств, предназначенных для поддержки отдельных этапов процесса.

В настоящее время существует очень много CASE средств и фирм, специализирующихся на их разработке (Rational). При выборе CASE средств следует руководствоваться основным принципом: сначала метод создания ПО, а потом – CASE средства, применимые для этого метода. Выбор наоборот чреват нехорошими последствиями.

**3. Стандартизации программной инженерии**

**Методологическую основу любой инженерии** составляет *понятие жизненного цикла (ЖЦ) изделия (продукта)* как совокупности всех действий, которые надо выполнить на протяжении всей «жизни» изделия.

**Весь процесс разработки ПО протекает в соответствии с принятыми на предприятии стандартами и нормативными документами**, регламентирующими требования к процессам жизненного цикла, качества, документирования программного продукта и оценке технологической зрелости организаций-разработчиков.

*Смысл жизненного цикла состоит во взаимосвязанности всех этих действий.* Жизненный цикл промышленного изделия определяется как последовательность этапов (фаз, стадий), состоящих из технологических процессов, действий и операций. Впервые о необходимости рассматривать разработку ПО с позиций его жизненного цикла заговорили в 1968 г. Исторически основные стандарты на ЖЦ ПО появились в 1985г.

*В мировой практике промышленного производства гарантией успеха являются стандарты на производство продуктов и услуг и сертификация производителей на соответствие этим стандартам*. Процесс стандартизации и сертификации давно вошел и в программную инженерию, где он составляет основу промышленного производства программных продуктов.

Среди всего многообразия стандартов принято выделять следующие основные **типы стандартов**:

**Корпоративные стандарты** разрабатываются крупными фирмами (корпорациями) с целью повышения качества своей продукции. Такие стандарты разрабатываются на основе собственного опыта и с учетом требований мировых стандартов. Корпоративные стандарты не сертифицируются, но являются обязательными для применения внутри корпорации. В условиях рыночной конкуренции могут иметь закрытый характер.

**Отраслевые стандарты** действуют в пределах организаций некоторой отрасли (министерства). Например, СНИП – строительные нормы и правила. Разрабатываются с учетом требований мирового опыта и специфики отрасли. Являются, как правило, обязательными для отрасли. Подлежат сертификации.

**Государственные стандарты (ГОСТы)** принимаются государственными органами, в некоторых случаях имеют силу закона. Разрабатываются с учетом мирового опыта или на основе отраслевых стандартов. Могут иметь как рекомендательный, так и обязательный характер (стандарты безопасности). Для сертификации создаются государственные или лицензированные органы сертификации.

**Международные стандарты.** Разрабатываются, как правило, специальными международными организациями на основе мирового опыта и лучших корпоративных стандартов. Имеют сугубо рекомендательный характер. ***Право сертификации получают организации (государственные и частные), прошедшие лицензирование в международных организациях.***

**Наиболее известными стандартами программной инженерии являются:**

* Международный стандарт ISO/IEC 12207 «Информационная технология. Жизненный цикл процессов разработки программного обеспечения». Стандарт содержит определения основных понятий программной инженерии. (в частности программного продукта и жизненного цикла программного продукта).
* ГОСТ Р ИСО/МЭК 12207-99 «Информационная технология. Процессы жизненного цикла программных средств».
* ГОСТ Р ИСО/МЭК 15288-2005 «Системная инженерия. Процессы жизненного цикла систем».
* ГОСТ Р ИСО/МЭК 15910-2002 «Процесс создания документации пользователя программного средства».
* ГОСТ Р ИСО/МЭК 9126-93 «Оценка программного продукта. Характеристики качества и руководящие указания по их применению».
* SEI CMM — Capability Maturity Model (for Software) — модель зрелости процессов разработки программного обеспечения. Стандарт отвечает на вопрос: «Какими признаками должна обладать профессиональная организация по разработке ПО?».
* **PMBOK** — **Project Management Body of Knowledge — Свод знаний по управлению проектами.**
* **SWЕBOK** — **Software Engineering Body of Knowledge — Свод знаний по программной инженерии** — содержит описания состава знаний по разделам (областям знаний) программной инженерии.
* ACM/IEEE CC2001 — Computing Curricula 2001 – Академический образовательный стандарт в области компьютерных наук. Выделены 4 основных раздела компьютерных наук: Computer science, Computer engineering, Software engineering и Information systems, по каждому из которых описаны области знаний соответствующего раздела, состав и планы рекомендуемых курсов.

Рассмотрим более подробно:

PMBoK - Свод знаний по [управлению проектами](https://ru.wikipedia.org/wiki/%D0%A3%D0%BF%D1%80%D0%B0%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D0%B5_%D0%BF%D1%80%D0%BE%D0%B5%D0%BA%D1%82%D0%B0%D0%BC%D0%B8)

SWEBOK (Software Engineering Body of Knowledge) — Области знаний программной инженерии

3.1 PMBoK - Свод знаний по [управлению проектами](https://ru.wikipedia.org/wiki/%D0%A3%D0%BF%D1%80%D0%B0%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D0%B5_%D0%BF%D1%80%D0%BE%D0%B5%D0%BA%D1%82%D0%B0%D0%BC%D0%B8)

Разработано несколько версийPMBoK - Свода знаний по [управлению проектами](https://ru.wikipedia.org/wiki/%D0%A3%D0%BF%D1%80%D0%B0%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D0%B5_%D0%BF%D1%80%D0%BE%D0%B5%D0%BA%D1%82%D0%B0%D0%BC%D0%B8) ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *Project Management Body of Knowledge*) представляет собой сумму профессиональных знаний по управлению проектами. [Институт управления проектами](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82_%D1%83%D0%BF%D1%80%D0%B0%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D1%8F_%D0%BF%D1%80%D0%BE%D0%B5%D0%BA%D1%82%D0%B0%D0%BC%D0%B8) использует этот документ в качестве основного справочного материала, руководства для своих программ по профессиональному развитию.

В *PMBoK*описываются суть процессов управления проектами в терминах интеграции между процессами и взаимодействий между ними, а также цели, которым они служат. Эти процессы разделены на пять групп, называемых «группы процессов управления проектом».

Согласно *PMBoK* ***Управление проектами***выполняется с помощью применения и интеграции логически сгруппированных 42 процесса управления проектами, объединенных в 5 групп процессов:

* инициация;
* планирование;
* исполнение;
* мониторинг и управление;
* завершение.

**В управление проектами, как правило, входит:**

1. определение требований;

2. удовлетворение различных потребностей, решение проблем и удовлетворение ожиданий различных заинтересованных сторон проекта в ходе планирования и выполнения проекта;

3. уравновешивание конкурирующих ограничений проекта, среди прочих: содержание; качество; расписание; бюджет; ресурсы; риски.

Блок-схема процессов (рисунок 2.3) кратко описывает основные зависимости и взаимодействия групп процессов и заинтересованных сторон проекта. Каждая группа процессов включает составляющие ее процессы управления проектами, которые связаны соответствующими входами и выходами, когда результат или выход одного процесса становится входом для другого.

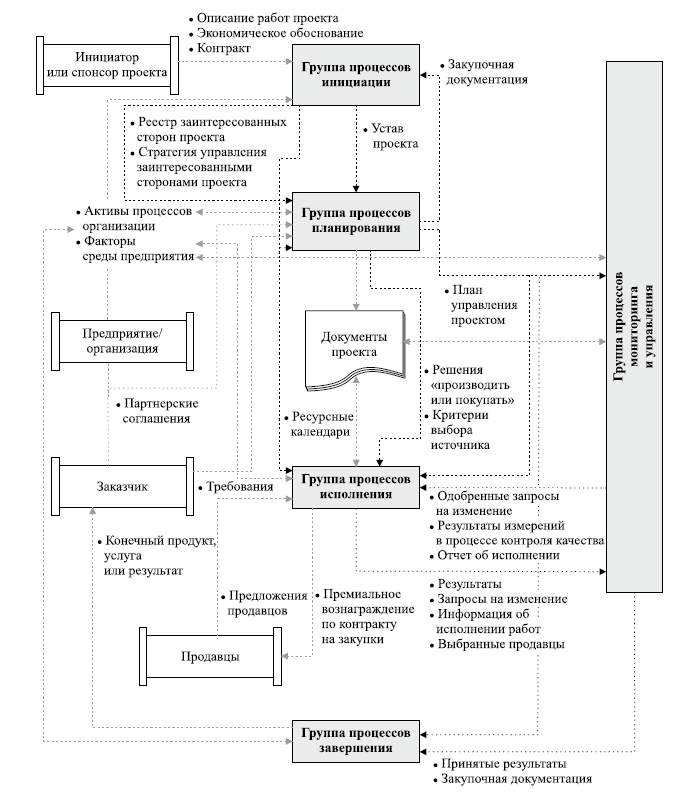


Рисунок 2.3. Взаимодействия процессов управления проектами

3.2 SWEBOK – Software Engineering Body of Knowledge (Свод знаний по программной инженерии)  ***-*** международный стандарт ISO/IEC TR 19759 от 2015 г, в котором описана общепринятая сумма знаний по программной инженерии

***SWEBOK*** *- документ, в котором описаны эталонные методики по всем стадиям разработки программного обеспечения.*

*SWEBOK является одним из самых авторитетных стандартов в области разработки программного обеспечения, постоянно переиздается и расширяется на предмет поэтапной детализации наиболее сложных и важных направлений работ.*

Согласно SWEBOK 2004, программная инженерия включает в себя 10 основных и 7 дополнительных областей знаний, на которых базируются процессы разработки ПО.

К **основным областям знаний** относятся следующие области (10):

* Software requirements — программные требования.
* Software design — дизайн (архитектура).
* Software construction — конструирование программного обеспечения.
* Software testing — тестирование.
* Software maintenance — эксплуатация (поддержка) программного обеспечения.
* Software configuration management — конфигурационное управление.
* Software engineering management — управление в программной инженерии.
* Software engineering process — процессы программной инженерии.
* Software engineering tools and methods — инструменты и методы.
* Software quality — качество программного обеспечения.

**Дополнительные области знаний**включают в себя:

* Computer engineering — разработка компьютеров.
* Computer science — информатика.
* Management — общий менеджмент.
* Mathematics — математика.
* Project management — управление проектами.
* Quality management — управление качеством.
* Systems engineering — системное проектирование.

Все это необходимо знать и уметь применять, для того чтобы разрабатывать ПО.

В настоящий момент сообществом разрабатывается новая, дополненная версия, включающая 15 областей, в дополнение к основным 10, добавляются еще 5:

* **software engineering professional practice** — описание критериев профессионализма и компетентности;
* **software engineering economics** — экономические аспекты разработки ПО;
* **computing foundations** — основы вычислительных технологий, применимых в разработке ПО;
* **mathematical foundations** — базовые математические концепции и понятия, применимые в разработке ПО;
* **engineering foundations** — основы инженерной деятельности.

**4 Модели жизненного цикла программных продуктов**

В общем случае модель жизненного цикла описывается в виде последовательности процессов, работ и задач, обеспечивающих разработку, эксплуатацию и сопровождение программного продукта и отражающих эволюцию изменения продукта, начиная от формулировки требований к ней до прекращения ее использования.

***Большинство существующих моделей жизненного цикла ПО являются разновидностями трех классических моделей:***

* + 1. каскадной (ступенчатой или водопадной);
    2. спиральной
    3. эволюционной (итеративной или инкрементальной);

**1. Каскадная модель (**классическая, водопадная, Waterfall**)**

*Каскадная модель* ([рисунок](https://www.intuit.ru/studies/courses/3590/832/lecture/31170?page=4#image.1.1) 2.4) разработки программного обеспечения является первой и, как следствие, самой критикуемой.

*Каскадная модель - это вид процесса разработки программного обеспечения, в которой процесс разработки выглядит как поток последовательно проходящих фаз анализа требований, проектирования, реализации, тестирования, интеграции и поддержки.*

Эта модель разработки была впервые презентована в 1970 году в статье Винстона Ройса. Он описал в виде концепции то, что сейчас принято называть "*каскадная модель*", и обсуждал ее недостатки. В этой же статье он показал, как этот тип процесса разработки программных продуктов может быть доработан до итеративной модели разработки *ПО*. В оригинальной каскадной модели Ройса следующие этапы процесса были представлены в таком порядке:

* Определение требований.
* Проектирование.
* Конструирование/реализация/кодирование.
* Воплощение.
* Тестирование/отладка/верификация.
* Инсталляция.
* Поддержка.

При работе с данной методологией предполагается последовательное выполнение этапов разработки. Такая структура не дает изменить требования к программному продукту до самого релиза.

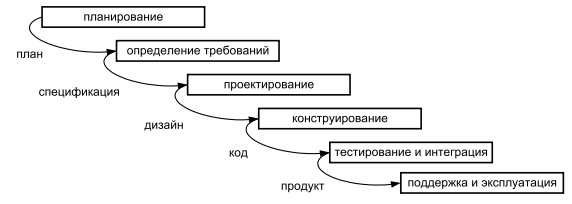


Рисунок 2.4. Каскадная модель разработки

В модели Waterfall легко управлять проектом. Благодаря её жесткости, разработка проходит быстро, стоимость и срок заранее определены. Но это палка о двух концах. Каскадная модель будет давать отличный результат только в проектах с [четко и заранее определенными требованиями](http://www.edsd.ru/ru/uslugi/poektirovanie) и способами их реализации.

Переход между фазами возможен только после полного и успешного завершения предыдущей. Следуя предложенной структуре, разработчик переходит от одной стадии к другой последовательно. Сначала полностью завершается первый этап ("*Определение* *требований*"), в результате которого появляется полный и исчерпывающий *список* требований к информационной системе. Только после этого возможен переход к проектированию, в ходе которого разрабатываются документы, подробно, ясно и непротиворечиво описывающие для программистов способ и план реализации зафиксированных требований и т. д.

*Недостатком этой модели является то, что в основу ее концепции положена модель фабрики, где продукт проходит стадии от замысла до производства, затем передается заказчику как готовое изделие, изменение которого не предусмотрено, хотя возможна замена на другое подобное изделие в случае рекламации или некоторых ее деталей, вышедших из строя.*

*Каскадная модель* строится на постулате последовательных переходов от одной фазы разработки к другой только после полного и успешного завершения предыдущей. Переходы назад, "перепрыжки" вперед, перекрытия фаз - недопустимы.

Каскадную модель организации процесса разработки программного обеспечения критикуют за недостаточную гибкость, объявление самоцелью формальное *управление проектом* в *ущерб* срокам, стоимости и качеству. Но ее четкая структурированность и формализация являются неоспоримыми ценностями, которые способствуют снижению многих возникающих рисков.

Но при этом для части проектов, которые затрагивают, к примеру, *безопасность* жизнедеятельности, строго поставленные требования и высокая степень формализации являются основополагающим и необходимым фактором.

**Когда использовать каскадную методологию?**

* Только тогда, когда требования известны, понятны и зафиксированы. Противоречивых требований не имеется.
* Нет проблем с доступностью программистов нужной квалификации.
* В относительно небольших проектах.

*В PMBOK 3-й версии формально была закреплена только методика каскадной модели и не были предложены альтернативные варианты, известные как итеративное ведение проектов или Agile.*

**2. Спиральная модель**

Следующей по времени возникновения стала *спиральная модель* (рисунки 2.5, 2.6). Спиральная модель основывается на итерационной процедуре использования каскадной модели.

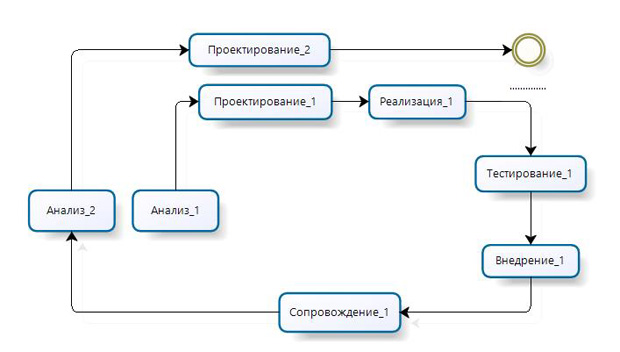
[](https://www.intuit.ru/EDI/18_05_18_2/1526595757-13793/tutorial/824/objects/1/files/01_02.jpg)

Рисунок 2.5**.**Спиральная модель разработки

**Набор фаз и их структура в спиральной модели соответствуют водопадной модели, но в спиральной методологии каждая фаза завершается *этапом прототипирования* и управления рисками.** Этап прототипирования после каждой фазы проекта позволяет определить, насколько текущее состояние проекта соответствует первоначальному плану. По итогам прототипирования выполняется либо переход к следующей фазе, либо возвращение на одну из предыдущих фаз для выполнения необходимых корректировок.



Рисунок 2.6. Спиральная модель разработки

Модель хорошо работает для решения критически важных бизнес-задач, когда неудача несовместима с деятельностью компании, в условиях выпуска новых продуктовых линеек, при необходимости научных исследований и практической апробации. Эта модель не подойдет для малых проектов, она резонна для сложных и дорогих, например, таких, как разработка системы документооборота для банка, когда каждый следующий шаг требует большего анализа для оценки последствий, чем программирование.

**3. Эволюционная модель (Итеративная и инкрементная модель)**

**Эволюционная модель** предполагает разбиение жизненного цикла проекта на последовательность итераций, каждая из которых предполагает создание фрагментов ПО меньшей функциональности, по сравнению с требованиями заказчика.

Цель каждой итерации — получение уже на ранних этапах разработки работающей версии программной продукции, содержащей определенную функциональность.

**С точки зрения структуры жизненного цикла** такую модель называют ***итеративной***, **с точки зрения развития продукта** — ***инкрементальной***.

В результате на каждой итерации можно анализировать промежуточные результаты работ и реакцию на них всех заинтересованных лиц и вносить корректирующие изменения на следующих итерациях. Каждая итерация может содержать полный набор принятых этапов ЖЦ: от анализа требований до ввода в эксплуатацию очередной части ПО.

Истоки концепции итеративной разработки прослеживаются в относящихся к 1930-м годам работах эксперта по проблемам качества продукции Уолтера Шеварта из Bell Labs. Важной вехой в истории является осуществленный в 50-е годы проект по разработке сверхзвукового реактивного самолета X-15. По мнению участников этих работ, применение спиральной методологии в значительной степени определило дальнейший успех проекта.

*Итеративный подход в разработке программного обеспечения - это выполнение работ параллельно с непрерывным анализом полученных результатов и корректировкой предыдущих этапов работы. Проект при этом подходе в каждой фазе развития проходит повторяющийся цикл PDCA: планирование - реализация - проверка - оценка.*

**Итерационная модель жизненного цикла** не требует для начала полной спецификации требований. Вместо этого, создание начинается с реализации части функционала, становящейся базой для определения дальнейших требований. Этот процесс повторяется. Версия может быть неидеальна, главное, чтобы она работала. Понимая конечную цель, мы стремимся к ней так, чтобы каждый шаг был результативен, а каждая версия — работоспособна.

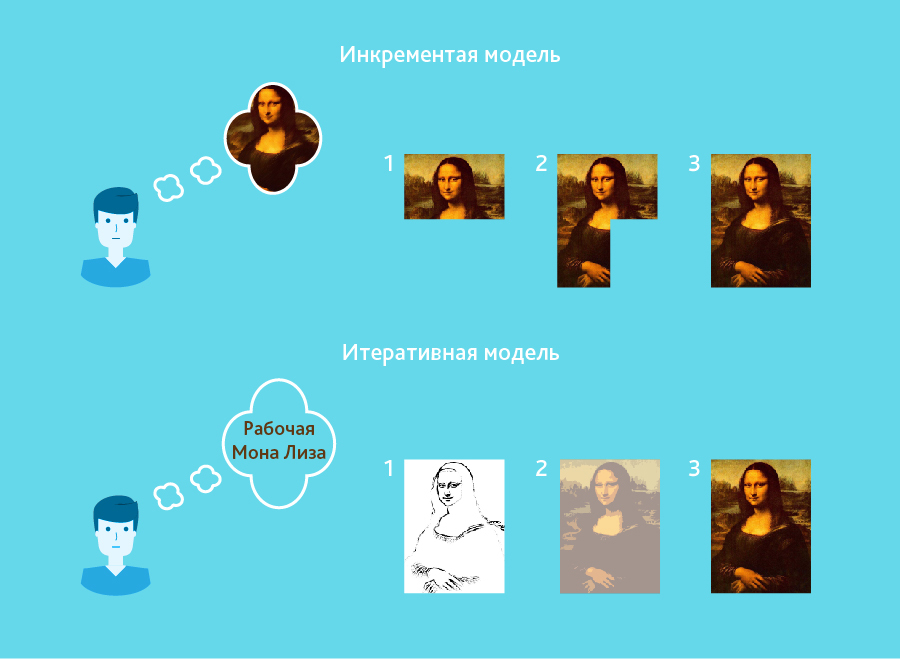


Рисунок 2.7.Итеративная и инкрементальная модели

В инкрементной модели функционал продукта (***продукт***) наращивается по кусочкам (Рисунок 7), продукт составляется из частей. В отличие от итерационной модели, каждый кусочек представляет собой целостный элемент.

При итерационной «разработке» - результат не определен (Мона Лиза). Как видно, в первой итерации есть лишь набросок Джоконды, во второй — появляются цвета, а третья итерация добавляет деталей, насыщенности и завершает процесс.

**Когда оптимально использовать итеративную модель?**

* Требования к конечной системе заранее четко определены и понятны.
* Проект большой или очень большой.
* Основная задача должна быть определена, но детали реализации могут эволюционировать с течением времени.

Итеративная модель, в противовес классической модели, не предполагает полного объема требований для начала *работ* *по* реализации продукта.

*С PMBOK 4-й версии был достигнут компромисс между приверженцами каскадной модели разработки ПО и профессионалами, делающими ставку на итеративные методы. Это привело к тому, что, начиная с 2009 года, Институтом управления проектами (PMI) предлагается как стандарт гибридный вариант методологии управления проектами, сочетающий в себе плюсы как от водопадной методологии, так и от достижения итеративных методов.*

Залог успешного применения этой модели - четко выстроенные этапы тестирования/отладки/верификации требований и тщательная *валидация* разрабатываемой функциональности в каждой из итераций.

Итеративная модель, *по* сути, является переходной (от каскадной к Agile) моделью разработки программного обеспечения и, *по* мнению многих специалистов, оптимальной моделью разработки программного обеспечения.

**«RAD Model» (rapid application development model или быстрая разработка приложений)**

RAD-модель — разновидность инкрементной модели. В RAD-модели компоненты или функции разрабатываются несколькими высококвалифицированными командами параллельно, будто несколько мини-проектов. Временные рамки одного цикла жестко ограничены. Созданные модули затем интегрируются в один рабочий прототип. Синергия позволяет очень быстро предоставить клиенту для обозрения что-то рабочее с целью получения обратной связи и внесения изменений (рисунок 2.8)



Рисунок 2.8.RAD-модель

Модель быстрой разработки приложений включает следующие фазы:

* Бизнес-моделирование: определение списка информационных потоков между различными подразделениями.
* Моделирование данных: информация, собранная на предыдущем этапе, используется для определения объектов и иных сущностей, необходимых для циркуляции информации.
* Моделирование процесса: информационные потоки связывают объекты для достижения целей разработки.
* Сборка приложения: используются средства автоматической сборки для преобразования моделей системы автоматического проектирования в код.
* Тестирование: тестируются новые компоненты и интерфейсы.

**Когда используется RAD-модель?**

Может использоваться только при наличии высококвалифицированных и узкоспециализированных архитекторов. Бюджет проекта большой, чтобы оплатить этих специалистов вместе со стоимостью готовых инструментов автоматизированной сборки. RAD-модель может быть выбрана при уверенном знании целевого бизнеса и необходимости срочного производства системы в течение 2-3 месяцев.

**5 Методологии процесса разработки ПО**

На текущий момент сложилось два основных направления.

**Первый и основной на сегодняшний день** - **последовательный подход,** состоящий в четком следовании *запланированной структуре* *работ*, суть которых определена заранее.

Адепты первого - это консерваторы, предпочитающие более основательные, "тяжелые" методологии, в которых четко и однозначно определена структура и последовательность выполняемых этапов *работ*. Типичным примером таких методологий является водопадная модель разработки информационных систем.

**Второй,** наиболее перспективный и амбициозный, - **гибкий подход,** являющийся последователем итерационного подхода, который не отрицает первого, но цель которого состоит в снижении сложности необходимой структуры этапов при разработке программного обеспечения, повышении прозрачности выполняемого набора *работ* и использовании творческой инициативы людей как средства повышения результативности. Такими методологиями являются процессы **семейства Agile.**

Приверженцы второго - это новаторы, выделившиеся из популярного ранее направления итерационной разработки, которые структуре и последовательности *работ* предпочитают эффективное профессиональное взаимодействие и гибкость в ответ на изменчивость внешнего мира.

Все семейство гибких методологий на сегодняшний день находятся в непрерывном развитии:

* модернизируются и дополняются существующие подходы;
* разрабатываются новые методологии;
* расширяется сфера их использования.

Методологии процессов разработки ПО принято классифицировать по «весу» — количеству формализованных процессов (большинство процессов или только основные) и детальности их регламентации. Чем больше процессов документировано, чем более детально они описаны, тем больше «вес» модели.

Наиболее распространенные современные модели процесса разработки ПО представлены на Рисунке 2.9.

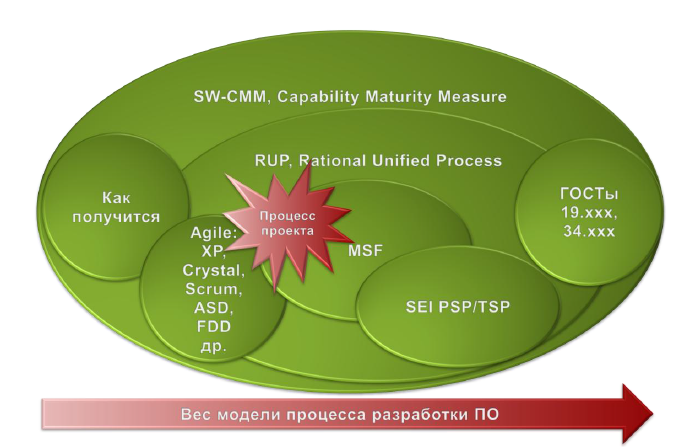


Рисунок 2.9. Методологии процесса разработки ПО и их распределение по «весу»

**5.1. Методологии Последовательного подхода**

#### ГОСТы

ГОСТ 19 «Единая система программной документации» и ГОСТ 34 «Стандарты на разработку и сопровождение автоматизированных систем» ориентированы на последовательный подход к разработке ПО. Разработка в соответствии с этими стандартами проводится по этапам, каждый из которых предполагает выполнение строго определенных работ, и завершается выпуском достаточно большого числа весьма формализованных и обширных документов. Таким образом, строгое следование этим Гостам не только приводит к водопадному подходу, но и требует очень высокой степени формализованности разработки. На основе этих стандартов разрабатываются программные системы по госзаказам в России.

#### SW-CMM

В середине 80-х годов минувшего столетия Министерство обороны США крепко задумалось о том, как выбирать разработчиков ПО при реализации крупномасштабных программных проектов. По заказу военных Институт программной инженерии, входящий в состав Университета Карнеги-Меллона, разработал SW-CMM, Capability Maturity Model for Software в качестве эталонной модели организации разработки программного обеспечения.  
Данная модель определяет пять уровней зрелости процесса разработки ПО.

* Начальный — процесс разработки носит хаотический характер. Определены лишь немногие из процессов, и успех проектов зависит от конкретных исполнителей.
* Повторяемый — установлены основные процессы управления проектами: отслеживание затрат, сроков и функциональности. Упорядочены некоторые процессы, необходимые для того, чтобы повторить предыдущие достижения на аналогичных проектах.
* Определенный — процессы разработки ПО и управления проектами описаны и внедрены в единую систему процессов компании. Во всех проектах используется стандартный для организации процесс разработки и поддержки программного обеспечения, адаптированный под конкретный проект.
* Управляемый — собираются детальные количественные данные по функционированию процессов разработки и качеству конечного продукта. Анализируется значение и динамика этих данных.
* Оптимизируемый — постоянное улучшение процессов основывается на количественных данных по процессам и на пробном внедрении новых идей и технологий.

Документация с полным описанием SW-CMM занимает около 500 страниц и определяет набор из 312 требований, которым должна соответствовать организация, если она планирует аттестоваться по этому стандарту на 5-ый уровень зрелости.

#### RUP

Унифицированный процесс (Rational Unified Process, RUP) был разработан Филиппом Крачтеном (Philippe Kruchten), Иваром Якобсоном (Ivar Jacobson) и другими сотрудниками компании "Rational Software" в качестве дополнения к языку моделирования UML. Модель RUP описывает абстрактный общий процесс, на основе которого организация или проектная команда должна создать конкретный специализированный процесс, ориентированный на ее потребности. Именно эта черта RUP вызывает основную критику — поскольку он может быть чем угодно, его нельзя считать ничем определенным. В результате такого общего построения RUP можно использовать и как основу для самого что ни на есть традиционного водопадного стиля разработки, так и в качестве гибкого процесса.

#### MSF

Microsoft Solutions Framework (MSF) — это гибкая и достаточно легковесная модель, построенная *на основе итеративной разработки*. Привлекательной особенностью MSF является большое внимание к созданию эффективной и небюрократизированной проектной команды. Для достижения этой цели MSF предлагает достаточно нестандартные подходы к организационной структуре, распределению ответственности и принципам взаимодействия внутри команды.

#### PSP/TSP

Одна из последних разработок Института программной инженерии Personal Software Process/ Team Software Process. Personal Software Process определяет требования к компетенциям разработчика. Согласно этой модели, каждый программист должен уметь:

* учитывать время, затраченное на работу над проектом;
* учитывать найденные дефекты;
* классифицировать типы дефектов;
* оценивать размер задачи;
* осуществлять систематический подход к описанию результатов тестирования;
* планировать программные задачи;
* распределять их по времени и составлять график работы.
* выполнять индивидуальную проверку проекта и архитектуры;
* осуществлять индивидуальную проверку кода;
* выполнять регрессионное тестирование.

Team Software Process делает ставку на самоуправляемые команды численностью 3-20 разработчиков. Команды должны:

* установить собственные цели;
* составить свой процесс и планы;
* отслеживать работу;
* поддерживать мотивацию и максимальную производительность.

Последовательное применение модели PSP/TSP позволяет сделать нормой в организации пятый уровень CMM.

## **5.2 Методологии подхода гибкой разработки Agile**

При сравнении классических (водопадной и итерационной) методик с гибкой методологией Agile необходимо осознание того, что *каскадная модель* - это подход хорошо описанный и детализированный, а Agile - это набор практик и принципов, в которых так или иначе поддерживаются различные методологии гибкой разработки проектов.

[Agile](https://www.atlassian.com/ru/agile) (agile software development, от англ. agile – проворный) — это структурированный итеративный подход к управлению проектами и разработке продуктов. В нем учитывается непостоянный характер разработки продуктов. Самоорганизующиеся команды, выбравшие этот подход, получают возможность реагировать на изменения, не отклоняясь от намеченного пути. В наши дни agile не дает особого преимущества перед конкурентами.

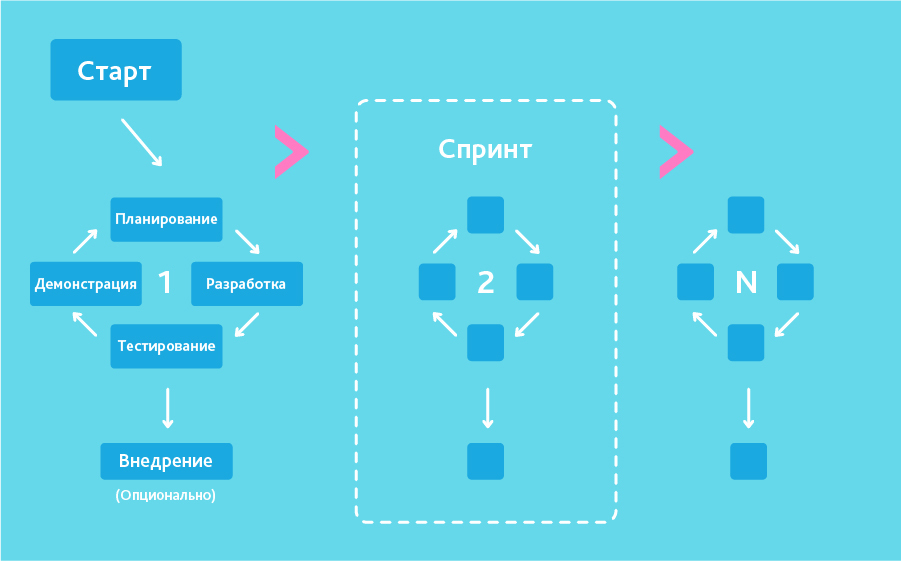


Рисунок 2.10.  Agile разработка

В «гибкой» методологии разработки после каждой итерации заказчик может наблюдать результат и понимать, удовлетворяет он его или нет. Это *одно из преимуществ гибкой модели*. К ее недостаткам относят то, что *из-за отсутствия конкретных формулировок результатов сложно оценить трудозатраты и стоимость, требуемые на разработку*.

**5.2.1) Экстремальное программирование (XP)**– это упрощенная методология организации разработки программ для небольших и средних по размеру команд разработчиков, занимающихся созданием программного продукта в условиях неясных или быстро меняющихся требований.

Основными целями XP являются **повышение доверия заказчика** к программному продукту путем предоставления реальных доказательств успешности развития процесса разработки и **резкое сокращение сроков разработки продукта**. При этом XP сосредоточено на минимизации ошибок на ранних стадиях разработки. Это позволяет добиться максимальной скорости выпуска готового продукта и даёт возможность говорить о прогнозируемости работы. Практически все приемы XP направлены на повышение качества программного продукта.

**5.2.2) Методология** [Kanban](https://www.atlassian.com/ru/agile/kanban)**это «подход баланса».**

Суть [Kanban](https://www.atlassian.com/ru/agile/kanban) — в визуализации работы, ограничении объема [незавершенной работы](https://www.atlassian.com/ru/agile/kanban/wip-limits) и достижении максимальной эффективности (или скорости). Задача [Kanban](https://www.atlassian.com/ru/agile/kanban) – сбалансировать разных специалистов внутри команды и избежать ситуации, когда дизайнеры работают сутками, а разработчики жалуются на отсутствие новых задач.

Вся команда едина – в **kanban** нет ролей владельца продукта и scrum-мастера. Бизнес-процесс делится не на универсальные спринты, а на стадии выполнения конкретных задач: «Планируется», «Разрабатывается», «Тестируется», «Завершено» и др.

Главный показатель эффективности в kanban – это среднее время прохождения задачи по доске. Задача прошла быстро – команда работала продуктивно и слаженно. Задача затянулась – надо думать, на каком этапе и почему возникли задержки и чью работу надо оптимизировать.

**Для визуализации agile-подходов используют доски: физические и электронные**. Они позволяют сделать рабочий процесс открытым и понятным для всех специалистов, что важно, когда у команды нет одного формального руководителя.

**5.2.3) Методология** [Scrum](https://www.atlassian.com/ru/agile/scrum)

[Scrum](https://www.atlassian.com/ru/agile/scrum) **– это «подход структуры»**. Над каждым проектом работает универсальная команда специалистов, к которой присоединяется еще два человека: владелец продукта и scrum-мастер. Первый соединяет команду с заказчиком и следит за развитием проекта (это не формальный руководитель команды, а скорее куратор). Второй помогает первому организовать бизнес-процесс: проводит общие собрания, решает бытовые проблемы, мотивирует команду и следит за соблюдением scrum-подхода.

Scrum-подход делит рабочий процесс на равные **спринты** – обычно это периоды от недели до месяца, в зависимости от проекта и команды. Перед спринтом формулируются задачи на данный спринт, в конце – обсуждаются  результаты, а команда начинает новый спринт. Спринты очень удобно сравнивать между собой, что позволяет управлять эффективностью работы.

В [Scrum](https://www.atlassian.com/ru/agile/scrum)высокий темп работы достигается за счет ее деления на спринты продолжительностью от одной до четырех недель с точными датами начала и окончания. Из-за узких временных рамок сложные задания приходится делить на более мелкие истории, и команда быстрее учится.

Ключевыми точками в спринте являются собрания по [планированию спринта](https://www.atlassian.com/ru/agile/scrum/sprint-planning) и по [обзору итогов спринта](https://www.atlassian.com/ru/agile/scrum/sprint-reviews), а также [ретроспективы](https://www.atlassian.com/ru/agile/scrum/retrospectives). Кроме того, в ходе спринта проходят ежедневные scrum-совещания (**[стендапы](https://www.atlassian.com/ru/agile/scrum/standups)**). Эти **[scrum-собрания](https://www.atlassian.com/ru/agile/scrum/ceremonies)**не отнимают много сил и проводятся на регулярной основе.

**Когда использовать Agile?**

* Когда потребности пользователей постоянно меняются в динамическом бизнесе.
* Изменения на Agile реализуются за меньшую цену из-за частых инкрементов.
* В отличие от модели водопада, в гибкой модели для старта проекта достаточно лишь небольшого планирования.

**5.3 Сравнение классических и гибкого подходов**

Сравнивая эти подходы (классические и гибкий), следует отметить, что оба имеют набор преимуществ и недостатков. Возможны ситуации, когда обоими методами можно организовать реализацию необходимого программного обеспечения, но на старте процессов разработки *входные данные* *по* ресурсным параметрам (*стоимость*, время, квалификация персонала), а также требования к качеству информационной системы и т.д. существенно влияют на выбор методологии.

В качестве ***сильных сторон*** **водопадной модели** следует выделить следующие:

* Легкость для понимания и последующего применения.
* Подробная структурированность, что облегчает ее применение к малоопытным командам.
* Модель с самого начала задает стабильные требования к проекту/продукту.
* Проекты легко контролируются, отслеживаются ресурсы, риски, время.
* Качество имеет первоочередной приоритет по сравнению со стоимостью и временем.

**Agile,** в свою *очередь*, обладает следующими ***сильными сторонами***:

* Итеративный подход к разработке программного обеспечения.
* Использование четких временных рамок.
* Заинтересованные пользователи вовлечены в процесс разработки с самого начала.
* Быстрое получение первой/пробной версии продукта для тестирования.
* Легко воспринимаются корректировки и изменения в процессе разработки.

***Слабыми сторонами подходов являются следующие:***

**"Каскад":**

* Требования должны быть определены и детально описаны до начала стадии разработки.
* Высокая цена.
* Медленный темп работы.
* Чувствительность к изменениям.
* Мало возможностей для конечного пользователя повлиять на цели проекта и требования к продукту.
* Зачастую проблемы выявляются только на этапе тестирования.
* Много документации, которая непонятна конечному пользователю или заказчику.

**Agile:**

* Может привести к низкому качеству продукта.
* Существует риск никогда не достигнуть поставленной цели при инициации процесса.
* Могут возникнуть проблемы с расширяемостью продукта.

На основе приведенных факторов становится возможным сделать целесообразные выводы о ситуациях, когда использование того или иного подхода является оптимальным.

Классические модели предпочтительнее использовать в ситуациях, когда требования к продукту предельно ясны и стабильны, определены используемые технологии и инструменты, речь идет о внедрении большого и сложного программного обеспечения. Примером может служить проекты внедрения ERP-систем.

Agile следует применять в тех случаях, когда *конечный пользователь* вовлечен в проект со старта, определены бизнес-цели проекта/продукта, проект небольшой или средний, относительно короткий *по* времени, состав команды стабильный, с высоким уровнем профессионализма, технические требования приемлемые.

При выборе методологии необходимо выбрать ту, которая подходит для достижения поставленных целей проекта. Необходимо понимать структуру, принципы, преимущества и недостатки каждой из них. В некоторых случаях это не выбор между методологиями, а правильная комбинация подходов для каждого из этапов конкретного процесса или проекта.

За последнее время изменился рынок потребителей *ПО*. Вместо больших проектов компании стремятся к использованию маленьких и средних проектов. Компании – разработчики *ПО* стремятся к более быстрому, частому, регулярному выпуску информационных систем.

Согласно недавнему опросу, 80% решений о внедрении в компании Agile-методологий и идей, принадлежат менеджерам высшего и среднего звена.

Итеративный и Agile методы разработки программных продуктов подходят не всем. Во всех компаниях существует своя специфика, которая определяется множеством разнообразных факторов различной природы, определяемой организационными, административными и специализированными условиями внешней и внутренней среды. В ряде компаний использование водопадной модели обосновано и экономически целесообразно. Можно привести много причин, почему Agile использовать не стоит.

**Резюме**

**Модели и методологии**

Waterfall, V, Инкременты, эволюция прототипов, спираль — ***модели***. Модели нужны чтобы предсказывать будущее поведение систем.

Scrum, Kanban, RUP, CMM, MSF, XP— ***методологии***, то есть набор артефактов и ритуалов. Методологии нужны чтобы система работала (и вообще существовала как система), а еще нужны чтобы продавать софт, консалтинг, книги и прочий инфобизнес.

Agile это не модель и не методология, это набор принципов на основе которых можно построить множество моделей и методологий.

По сути имеются всего два больших подхода:

* «пытаемся представить систему целиком в деталях» = waterfall, RUP/OpenUP etc
* «понимаем что всё систему представить пока не можем, начинаем с чего попроще» = всё гибкое и «быстрое» (хотя «быстрость» не гарантирует что полноценный продукт появится раньше, чем его бы делали по водопаду)

*Недостатки методов - Субъективно :)*

1. каскадная: требования не меняются в процессе разработки
2. V-Model: задачи разбиваются на подзадачи, для каждой задачи создается задача проверки
3. инкрементная: имплементят фичи одну за другой, пока их становится слишком много
4. RAD: рисование UML и прочих диаграмм, генерация кода по ним, SQL в визуальном редакторе, рисование формочек в конструкторе и т.д.
5. Agile: ежедневные митинги, где можно похвастаться достижениями и пожаловаться на проблемы
6. итеративная: быстро делают протототип, потом долго отлавливают баги, отвлекаясь на изменяющиеся требования заказчика
7. спиральная: в колесо цикла разработки вставляется палка в виде анализа рисков

***Основная литература***

1. Архипенков С. Лекции по управлению программными проектами// Электронный ресурс:http://citforum.ru/SE/project/

2. Основы управления ИТ-проектами [Электронный ресурс]: учебное пособие / сост. Е.Р. Кирколуп, Ю.Г., Скурыдин, Е.М. Скурыдина. – Барнаул : АлтГПУ, 2017// Электронный ресурс:http://library.altspu.ru/dc/pdf/kirkolup.pdf

3. Рыбалова Е.А. Управление проектами: учебное пособие / Е.А. Рыбалова. - Томск: ФДО, ТУСУР, 2015. - 205 с.

4. Бухарев Н.Р. Введение в гибкое управление проектами. Краткий конспект лекций// tulpar.kfu.ru/course/view.php?id=2284

[5. Коптелов А., Никитин И., Цулая М. Гибкая процессная методология Agile](https://www.intuit.ru/studies/courses/3590/832/info). - М.: ИНТУИТ // Электронный ресурс: // www.intuit.ru/studies/courses/3590/832/print\_lecture/31170

6. Топ-7 методов управления проектами: Agile, Scrum, Kanban, PRINCE2 и другие// www.pmservices.ru/project-management-news/top-7-metodov-upravleniya-proektami-agile-scrum-kanban-prince2-i-drugie/

### 7.[Грекул](https://www.intuit.ru/intuituser/userpage/v_i_grekul) В., [Коровкина](https://www.intuit.ru/intuituser/userpage/n_l_korovkina) Н., [Куприянов](https://www.intuit.ru/intuituser/userpage/y_v_kupriyanov) Ю. [Методические основы управления ИТ-проектами](https://www.intuit.ru/studies/courses/646/502/info). - М.: ИНТУИТ // Электронный ресурс: https://www.intuit.ru/studies/courses/646/502/info

### ***Дополнительная литература:***

8. Сериков А. Основы управления проектами. - М.: ИНТУИТ. Электронный ресурс: www.intuit.ru/studies/professional\_retraining/965/courses/272/info

9. Управление проектами: Учебное пособие / М.В. Романова. - М.: ИД ФОРУМ: НИЦ ИНФРА-М, 2014

10. Руководство к Своду знаний по управлению проектами РМВОК (A Guide to the Project Management Body of Knowledge PMBOK®) — 4-е изд. — Project Management Institute, Inc., 2008. — 464 с.

11. https://products.office.com/

Лекция 3. Управление проектами: определения и концепции

*План:*

1. Определение проекта

2 Системная модель управления проектами

2.1 Объекты управления

2.2 Функциональные области управления

2.3 Субъекты управления.

2.4 Команда управления проектом

2.5 Стейкхолдеры и организационные структуры

2.6 Организационная структура компании

**1. Определение проекта**

Существует ряд определений термина «проект», каждое из которых имеет право на существование, в зависимости от конкретной задачи, стоящей перед специалистом. Приведем некоторые определения проекта, взаимно дополняющие друг друга.

***Проект*** — *это комплексное, не повторяющееся мероприятие, предполагающее внедрение нового, ограниченное по времени, бюджету, ресурсам, а также четкими указаниями по выполнению, разработанными под потребности заказчика* [2].

***Проект***— *это ограниченное по времени, целенаправленное изменение отдельной системы с установленными требованиями к качеству результатов, с ограничениями расходования средств и со специфической организацией* (PMI).

***Проект*** *(лат. projectus — «брошенный вперед») — это временное предприятие, предназначенное для создания уникальных продуктов, услуг или результатов* (Руководство PMBOK® Guide — 2008 Edition, издание 4) [7].

***Проект***— *это*:

1) *предприятие, которое характеризуется принципиальной уникальностью условий его деятельности, таких как цели, время, затраты, качественные характеристики и другие условия, и отличается от других подобных предприятий специфической* ***проектной*** *организацией;*

2) *уникальный набор скоординированных действий с определенным началом и завершением, осуществляемых индивидуумом или организацией для решения специфических задач с определенным расписанием, затратами и параметрами выполнения* (Организационно-деятельностная модель ICB IPMA).

***Проект***— *это уникальный процесс, состоящий из набора взаимоувязанных и контролируемых работ с датами начала и окончания и предпринятый, чтобы достичь цели соответствия конкретным требованиям, включая ограничения по времени, затратам и ресурсам* (Процессная модель ISO 9000, 10006).

***Проект***— *целенаправленная деятельность временного характера, предназначенная для создания уникального продукта или услуги, ограниченная во времени и связанная с потреблением ресурсов* (Национальные требования к компетенции СОВНЕТ).

***Проект***— *это*:

1) *совокупность документов (расчетов, чертежей и т. д.) для создания какого-либо сооружения или изделия;*

2) *предварительный текст какого-либо документа;*

3) *замысел, план* (Советский энциклопедический словарь).

***Проект***— *это временное предприятие, предназначенное для создания уникальных продуктов, услуг или результатов* (PMBOK PMI).

Временный характер проекта означает, что у любого проекта есть определенное начало и завершение. Завершение наступает, когда достигнуты цели проекта или признано, что цели проекта не будут или не могут быть достигнуты, или исчезла необходимость в проекте. Каждый проект приводит к созданию уникального продукта, услуги или результата.

Из данных определений следует, что всем проектам присуще две важные характеристики.

1.Наличие дат начала и завершения (у каждого проекта есть начало и конец, этим проектная деятельность отличается от операционной, рутинной деятельности).

2.Результат каждого проекта – уникальный продукт или новая услуга. Это ещё одно отличие проектной деятельности от операционной, в процессе которой серийно производится определённая продукция или оказывается постоянная определённая услуга.

*Примерами проектов могут служить:*

• разработка нового продукта или услуги;

• осуществление изменений в структуре, кадрах и стиле организации;

• разработка или приобретение новой или усовершенствованной информационной системы;

• строительство здания или сооружения;

• внедрение новой процедуры или нового процесса на предприятии.

**Проект - это средство стратегического развития** (рисунок 3.1). Цель – описание того, что мы хотим достичь. Стратегия – констатация того, каким образом мы собираемся эти цели достигать. Проекты преобразуют стратегии в действия, а цели в реальность.



Рисунок 3.1. Проект – средство стратегического развития

Таким образом, каждая работа, которую выполняет конкретный сотрудник, привязывается к достижению стратегических целей организации.

Проекты объединяются в программы. *Программа* - ряд связанных друг с другом проектов, управление которыми координируется для достижения преимуществ и степени управляемости, недоступных при управлении ими по отдельности.

Проекты и программы объединяются в портфели. *Портфель* - набор проектов или программ и других работ, объединенных вместе с целью эффективного управления данными работами для достижения стратегических целей.

Проекты и управление ими существовали всегда. В качестве самостоятельной области знаний управление проектами начало формироваться в начале ХХ века. В этой дисциплине пока нет единых международных стандартов. Наиболее известные центры компетенции:

■ PMI, Project Management Institute, PMBOK — американский национальный стандарт ANSI/PMI 99-001-2004.  
■ IPMA, International Project Management Association. В России — СОВНЕТ.

***Основными признаками проекта являются***:

1) новизна;

2) изменения как основное содержание проекта;

3) неповторимость;

4) конкретная цель, ограниченная во времени;

5) временная ограниченность продолжительности проекта;

6) ограниченность требуемых ресурсов;

7) бюджет, относящийся к проекту;

8) комплексность решения проблемы;

9) уникальность конечных продуктов, технологии;

10) выделение сферы проекта в сфере взаимодействия организации и рынка.

В качестве примеров приведем такие проекты, как строительство микрорайона с созданием необходимой инфраструктуры, оптимизация энергопотребления области, создание и обеспечение выпуска нового автомобиля, модернизация предприятия, реорганизация коммунального хозяйства города, внедрение на предприятии международной системы управления качеством *IS0 9000*.

***Управление проектами (УП)*** — *это приложение знаний, навыков, инструментов и методов к работам проекта для удовлетворения требований, предъявляемых к проекту* (РМВОK PMI, издание 4)

[Руководство к Своду знаний по управлению проектами РМВОК (A Guide to the Project Management Body of Knowledge PMBOK®) — 4-е изд. — Project Management Institute, Inc., 2008. — 464 с].

***Управление проектами*** — *профессиональная творческая деятельность по руководству людскими и материальными ресурсами путем применения современных методов, средств и искусства управления для успешного достижения заранее поставленных целей при определенных требованиях к срокам, бюджету и характеристикам ожидаемых результатов проектов, осуществляемых в рыночных условиях в социальных системах*

[Туккель И. Л. Управление инновационными проектами: учебник для вузов / И. Л. Туккель, А. В. Сурина, Н. Б. Культин / под ред. И. Д. Туккеля. — СПб: БХВ-Петербург, 2014. — 416 с.

Управление инвестиционно-строительными проектами: международный подход : руководство / И.И.Мазур [и др.] ; под ред. И.И.Мазура, В. Д.Шапиро. — 2-е изд., перераб. —М. : Омега-Л, 2010. — 736 с.].

**Процесс** — это набор взаимосвязанных действий и операций, осуществляемых для получения заранее определенного продукта, результата или услуги. Каждый процесс характеризуется своими входами, инструментами и методами, которые могут быть применены, и конечными выходами (PMBOK, издание 3).

**В систему управления проектами** включаются такие элементы, как:

• субъекты управления проектами, к которым относятся внешние и внутренние участники проекта;

• объект управления, в качестве которого рассматривается сам проект;

• процессы управления, к которым относят процессы инициации, планирования, исполнения, контроля и завершения.

*Процессы управления проектом -*  обеспечивают результативное выполнение проекта в течение всего времени его существования.

***Процессы управления проектом*** подробно описываются стандартом РМВОK

**Свод знаний по**[**управлению проектами**](https://ru.wikipedia.org/wiki/%D0%A3%D0%BF%D1%80%D0%B0%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D0%B5_%D0%BF%D1%80%D0%BE%D0%B5%D0%BA%D1%82%D0%B0%D0%BC%D0%B8) ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *Project Management Body of Knowledge, PMBoK*), который представляет собой сумму профессиональных знаний по управлению проектами, признанных в качестве стандарта.

***Стандарт*** — *официальный документ, в котором описываются установленные нормы, методы, процессы и практики.*

В настоящем руководстве описываются суть процессов управления проектами в терминах интеграции между процессами и взаимодействий между ними, а также цели, которым они служат. Эти процессы разделены на пять групп, называемых «группы процессов управления проектом».

**Ограничения в проектах**

Под  ограничениями или “**железном треугольнике**” [PMI PMBOK3, 2004, Рус, с.8] подразумевают:

1. Содержание проекта

2. Время

3. Стоимость

**Задача проекта** — достижение конкретной бизнес-цели, при соблюдении ограничений «железного треугольника» (рисунок 3.2). Это означает, что ни один из углов треугольника не может быть изменен без оказания влияния на другие. Например, чтобы уменьшить время, потребуется увеличить стоимость и/или сократить содержание.

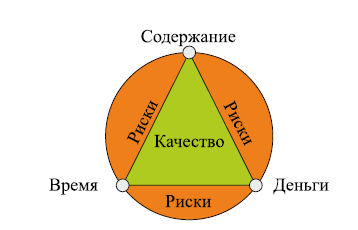
**

Рисунок 3.2. Проектный треугольник

Например, изменение *содержания* обычно влияет на *стоимость* проекта, но может и повлиять на *план* коммуникаций или *качество* продукта.

Согласно текущей редакции стандарта PMBOK, **проект считается успешным, если удовлетворены все требования заказчика и участников проекта**. Поэтому у проекта разработки ПО сегодня не три, а четыре фактора успеха:

* Выполнен в соответствие со спецификациями.
* Выполнен в срок.
* Выполнен в пределах бюджета.
* Каждый участник команды уходил с работы в 18:00 с чувством успеха.

Этот четвертый фактор успеха должен стать воспроизводимым, если предприятие хочет быть эффективным. Для успешного проекта характерно постоянное ощущение его участниками чувства удовлетворения и гордости за результаты своей работы, чувства оптимизма. Нет ничего более гибельного для проекта, чем равнодушие или уныние его участников.

**Система ограничений** может строиться на основе приоритетов проекта и должна учитывать требования потребителей к создаваемому продукту или услуге (Рисунок 3.3).

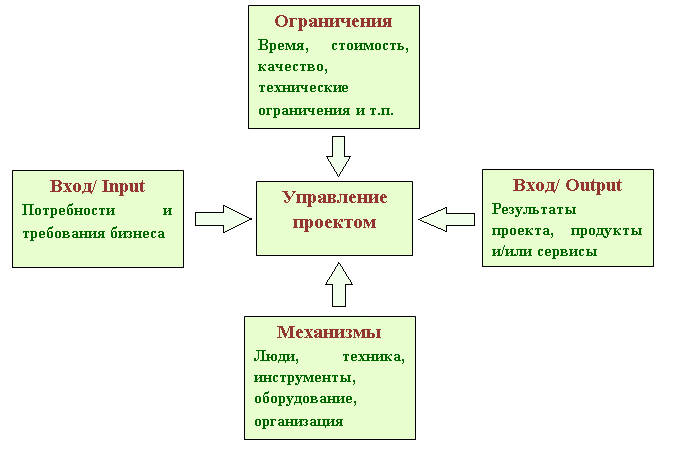


Рисунок 3.3. Процесс управления проектом. Роль ограничений. Источник: [APM PMBoK, 2000, c.15]

**2 Системная модель управления проектами**

*Системная модель управления проектами* представлена на рисунке 3.4.

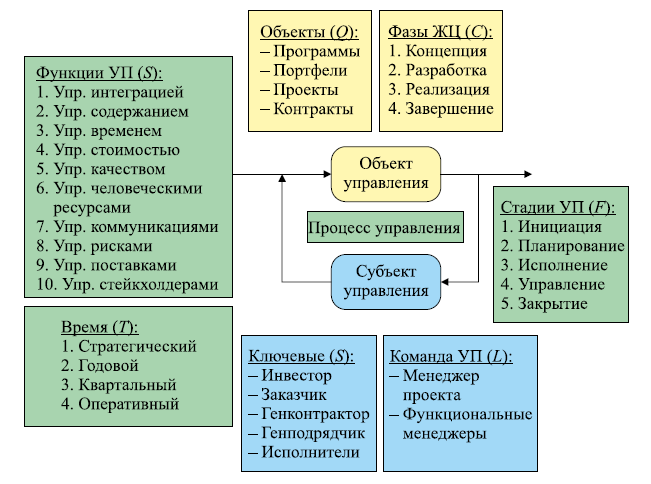


Рисунок 3.4. Системная модель управления проектами

В качестве методологической основы для определения и разработки задач, необходимых при управлении проектами, используются *основания* системной методологической модели управления проектами:

• Субъекты управления.

• Команда управления проектом.

• Виды или категории объектов управления (проектно-ориентированной деятельности и/или проектов).

• Объекты управления — проекты, программы, портфели проектов и программ, проектно-ориентированная деятельность в организации или множестве организаций.

• Фазы жизненного цикла объектов управления.

• Уровни управления.

• Функциональные области управления.

• Стадии процесса управления.

***2.1 Объекты управления***.

Объектами системы управления могут быть: программы, проекты, контракты (проекты), реализуемые в организациях или предприятиях, фазы жизненного цикла объекта управления: концепция, разработка, реализация, завершение.

***2.2 Функциональные области управления***.

**С точки зрения временного разреза управления проектом существуют следующие уровни управления**:

• *стратегический уровень*, который охватывает весь жизненный цикл проекта и соответствует организационно-экономическому уровню проекта;

• *годовой и квартальный уровни* управления, рассматривающие работы проекта, выполнение которых запланировано в течение года и квартала соответственно;

• *оперативный уровень* управления, занимающийся работами проекта, выполнение которых запланировано в течение месяца, декады, недели, суток, смены и т. д.

**Функции или *области управления* в проекте включают управление: интеграцией проекта; замыслом и работами; временными параметрами; стоимостью; качеством; рисками; персоналом; коммуникациями; контрактами или поставками**. Кроме этого предлагается дополнительно рассматривать такие области, как:

* управление изменениями в проекте;
* управление конфликтами;
* управление безопасностью проекта.

**Организационные уровни управления** (портфель, программа, проект)

***Портфель*** — *это набор проектов или программ и других работ, объединенных вместе с целью эффективного управления данными работами для достижения стратегических целей.*

***Управление портфелями*** относится к централизованному управлению одним или несколькими портфелями, что включает выявление, установление приоритетов, авторизацию, управление и контроль проектов, программ и других связанных работ с целью достижения определенных стратегических целей. Управление портфелями предусматривает обеспечение пересмотра проектов и программ с целью установления приоритетов при распределении ресурсов и соответствия портфеля стратегиям организации.

***Программа*** — *это ряд связанных друг с другом проектов, управление которыми координируется для достижения преимуществ и степени управляемости, недоступных при управлении ими по отдельности.*

Программы могут содержать элементы работ, имеющих к ним отношение, но лежащих за пределами содержания отдельных проектов программы. Проект может быть или не быть частью программы, но программа всегда содержит проекты.

***Управление программой*** определяется как централизованное, скоординированное управление группой проектов для достижения стратегических целей и преимуществ программы.

Проекты в рамках программ связаны посредством общего результата или совместных функциональных возможностей. Если связь между проектами заключается только в наличии общего клиента, продавца, технологии или ресурса, то предпринимаемыми действиями следует управлять как портфелем проектов, а не программой.

Проекты, входящие в программы или портфели, являются средствами достижения целей и задач организации, зачастую в контексте стратегического плана.

***2.3 Субъекты управления.***

Субъекты управления проектами — это те, кто управляет объектами управления. Субъектами проектного управления могут являться сотрудники, подразделения компании, а также коллегиальные (координационные советы, управляющие советы и т.д.) и временные органы управления (проектные группы) (рисунок 3.5).



Рисунок 3.5. Основные действующие лица проекта

***2.4 Команда управления проектом.***

Участники проекта (программы) — это субъекты управления, активно взаимодействующие между собой и с объектом управления при выработке и принятии управленческих решений в процессе его осуществления.

*К основным субъектам управления проектом* относятся:

1. Ключевые участники проекта (инвестор, заказчик, генконтрактор, генподрядчик, исполнители) показаны на рисунке 3.6.

2. Команда управления проектом: менеджер проекта, функциональные менеджеры проекта — члены команды проекта.

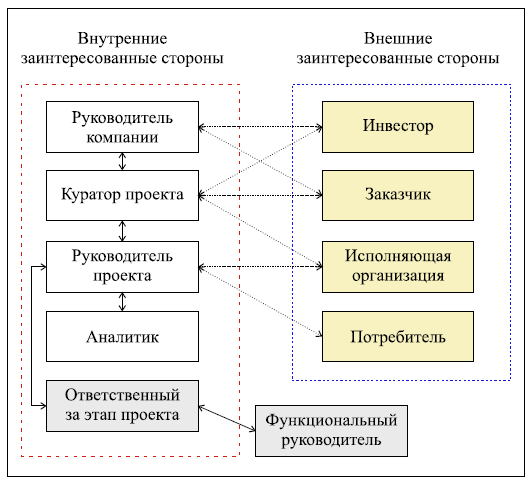
****

Рисунок 3.6. Ключевые участники проекта

Многие менеджеры проектов сосредоточиваются на «технических» ролях, таких как проектировщики баз данных, специалисты по сетям, эксперты по пользовательскому интерфейсу и т. д. Все они важны, но нужно подумать и о ролях «психологического» плана, которые могут играть один или более участников команды.

На укрупненном уровне роли, выполняемые участниками проектной команды, можно подразделить на три группы:

• роли, ориентированные на выполнение задач команды;

• роли, ориентированные на создание/поддержание работы команды;

• индивидуальные роли (нефункциональные).

Для того чтобы команда работала эффективно, одинаково важны роли первой и второй групп (рисунок 3.7).

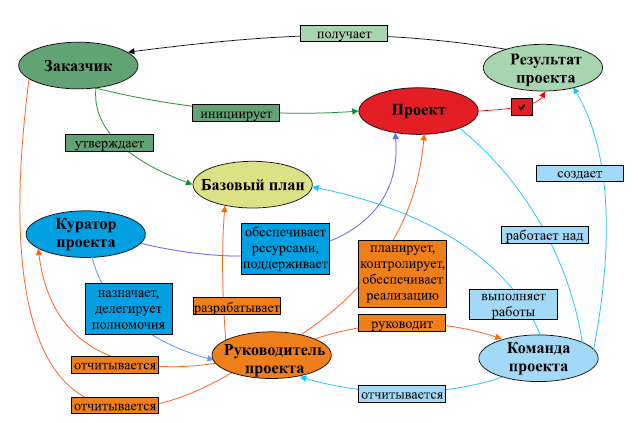
****

Рисунок 3.7. Определения проектного менеджмента и их взаимосвязи

**Команда проекта** — специфическая организационная структура, возглавляемая руководителем проекта и создаваемая на период осуществления проекта с целью эффективного достижения его целей.

Чем меньше проект, тем больше ролей приходится совмещать одному исполнителю***.***

***Роли и ответственности участников типового проекта разработки ПО*** можно условно разделить на пять групп:

1. Анализ. Извлечение, документирование и сопровождение требований к продукту.

2. Управление. Определение и управление производственными процессами.

3. Производство. Проектирование и разработка ПО.

4. Тестирование. Тестирование ПО.

5. Обеспечение. Производство дополнительных продуктов и услуг.

***Группа анализа*** включает в себя следующие роли:

* Бизнес-аналитик. Построение модели предметной области (онтологии).
* Бизнес-архитектор. Разрабатывает бизнес-концепцию системы. Определяет общее видение продукта, его интерфейсы, поведение и ограничения.
* Системный аналитик. Отвечает за перевод требований к продукту в функциональные требования к ПО.
* Специалист по требованиям. Документирование и сопровождение требований к продукту.
* Менеджер продукта (функциональный заказчик). Представляет в проекте интересы пользователей продукта.

***Группа управления*** состоит из следующих ролей:

 Руководитель проекта. Отвечает за достижение целей проекта при заданных ограничениях (по срокам, бюджету и содержанию), осуществляет операционное управление проектом и выделенными ресурсами.

 Куратор проекта. Оценка планов и исполнения проекта. Выделение ресурсов.

 Системный архитектор. Разработка технической концепции системы. Принятие ключевых проектных решений относительно внутреннего устройства программной системы и ее технических интерфейсов.

 Руководитель группы тестирования. Определение целей и стратегии тестирования, управление тестированием.

 Ответственный за управление изменениями, конфигурациями, за сборку и поставку программного продукта.

***В производственную группу*** входят:

 Проектировщик. Проектирование компонентов и подсистем в соответствие с общей архитектурой, разработка архитектурно значимых модулей.

 Проектировщик базы данных.

 Проектировщик интерфейса пользователя.

 Разработчик. Проектирование, реализация и отладка отдельных модулей системы.

***В большом проекте*** может быть несколько производственных групп, ответственных за отдельные подсистемы. Как правило, проектировщик выполняет роль лидера группы и управляет своим подпроектом или пакетом работ. Стоит не забывать, что руководитель проекта делегирует полномочия, но не ответственность.

***Группа тестирования*** в проекте состоит из следующих ролей:

 Проектировщик тестов. Разработка тестовых сценариев.

 Разработчик автоматизированных тестов.

 Тестировщик. Тестирование продукта. Анализ и документирование результатов.

***Участники группы обеспечения***, как правило, не входят в команду проекта. Они выполняют работы в рамках своей процессной деятельности. К группе обеспечения можно отнести следующие проектные роли:

 Технический писатель.

 Переводчик.

 Дизайнер графического интерфейса.

 Разработчик учебных курсов, тренер.

 Участник рецензирования.

 Продажи и маркетинг.

 Системный администратор.

 Технолог.

 Специалист по инструментальным средствам.

 Другие.

В зависимости от масштаба проекта одну роль могут исполнять несколько человек. Например, разработчики, тестировщики, технические писатели. Некоторые роли всегда должен исполнять только один человек. Например, Руководитель проекта, Системный архитектор. Один человек может исполнять несколько ролей. Возможны следующие совмещения ролей:

 Руководитель проекта + системный аналитик (+ системный архитектор)

 Системный архитектор + разработчик

 Системный аналитик + проектировщик тестов (+ технический писатель)

 Системный аналитик + проектировщик интерфейса пользователя

 Ответственный за управление конфигурациями + ответственный за сборку и поставку (+ разработчик)

**2.5 Стейкхолдеры и организационные структуры**

Факторы среды предприятия охватывают как внутренние, так и внешние факторы среды, окружающие проект или влияющие на его успех. Эти факторы могут возникать со стороны любого или всех предприятий, вовлеченных в проект. Факторы среды предприятия могут расширить или ограничить возможности управления проектом, а также положительно или отрицательно сказаться на результате (рисунок 3.8).

В большинстве процессов планирования такие факторы рассматриваются как входы. **К факторам среды предприятия** относятся (перечень не исчерпывающий):

• организационная культура, структура и процессы, инфраструктура (существующие сооружения и основное оборудование);

• государственные и промышленные стандарты (предписания контролирующих органов, кодексы поведения, стандарты на продукцию, стандарты качества, стандарты изготовления);

• имеющиеся человеческие ресурсы (навыки, знания, специализации, такие как проектирование, разработка, юридические вопросы, заключение контрактов и закупки);

• управление персоналом (правила приема на работу и увольнения, оценка эффективности работы и обучение персонала, правила сверхурочной работы и учет рабочего времени);

• корпоративная система авторизации работ, ситуация на рынке, готовность к риску заинтересованных сторон проекта, политический климат;

• каналы коммуникаций, принятые в организации; коммерческие базы данных (стандартизированные сметные данные, данные изучения промышленных рисков и базы данных рисков);

• информационные системы управления проектами (программное обеспечение для управления расписанием, система управления конфигурацией, система сбора и распространения информации и веб-интерфейсы к другим автоматизированным системам, работающим в режиме онлайн).

****

Рисунок 3.8. Заинтересованные лица проекта (стейкхолдеры)

К *факторам* ***ближнего окружения*** относят:

• руководство предприятия (определяет цели и основные требования к проекту), сферу финансов (определяет бюджетные рамки, способы и источники финансирования);

• сферу сбыта (формирует важные требования и условия к проекту, связанные с рынком сбыта, поведением покупателей и действием конкурентов);

• сферу производства (связана с рынком средств производства, определяет выбор технологии, оптимизацию мощностей и затрат);

• сферу материального обеспечения (связана с рынком сырья и полуфабрикатов и формирует требования к обеспечению сырьем, материалами по приемлемым ценам);

• сферу инфраструктуры (связана с рынком услуг и выдвигает требования к рекламе, транспорту, связи, информационному и прочему обеспечению).

*Факторами* ***внешнего окружения*** являются:

• политические условия (политическая стабильность, поддержка проекта правительством, уровень преступности);

• экономические факторы (тарифы и налоги, уровень инфляции и стабильность валюты, банковская система);

• правовые условия (правовое и законодательное обеспечение инвестиционной деятельности), социальные, природные условия;

• инфраструктура (наличие и стоимость сырья, воды, энергии, сбытовая сеть, логистика, уровень конкуренции и пр.).

***2.6 Организационная структура компании***

***Организационная структура***является фактором среды предприятия, оказывающим влияние на доступность ресурсов и на выполнение проектов.

Организационная структура компании отражает ее внутреннее устройство, потоки управляющих воздействий, распределение труда и специфические особенности производства. Организационные структуры варьируются от *функциональных* до *проектных*, при этом между ними существует множество матричных структур.

***1) Функциональная структура*** организации является иерархической (рисунок 3.9), в которой *у каждого сотрудника есть один явный вышестоящий начальник. Штатные сотрудники сгруппированы по специальностям, таким как производство, маркетинг, технические специальности и бухгалтерский учет, на высшем уровне.* Далее специальности могут подразделяться на функциональные подразделения, такие как машиностроение и электротехника. Каждый отдел функционального подразделения будет выполнять свою работу по проекту независимо от других отделов.

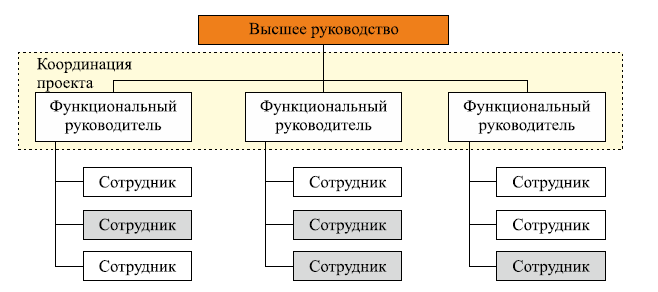


Рисунок 3.9. Функциональная структура

Функциональная структура имеет следующие особенности:

 Сохраняется принцип единоначалия

 Понятные и стабильные условия работы

 Хорошо приспособлены для операционной деятельности.

 Специализация подразделений позволяет накапливать экспертизу.

 Затруднено принятие решений и коммуникации между исполнителями. Осуществляются только через руководство.

 Управление сконцентрировано и держится на компетенции высшего руководства

Как правило, неэффективен контроль за ходом проекта (нет целостной картины)

Функциональная структура предполагает **многоуровневую иерархию**. Руководители функциональных подразделений это начальники управлений, начальники подчиненных им служб, отделов, лабораторий, секторов, групп. А еще у каждого начальника есть заместитель и, порой, не один. Примеры: министерства, ведомства, научные институты и предприятия советского периода.

***2) Матричные организации***, как показано на рисунках 3.10-3.12, представляют собой сочетание функциональных и проектных характеристик.

*Слабые матрицы* сохраняют многие из характеристик функциональной организации, а роль менеджера проекта больше напоминает роль координатора или диспетчера, нежели роль фактического менеджера проекта.

*Сильные матрицы* обладают многими характеристиками проектной организации и могут иметь менеджеров проектов с полной занятостью, имеющих существенные полномочия, а также административный персонал проекта, занятый полный рабочий день. Хотя сбалансированная матричная организация и признает необходимость существования менеджера проекта, она не наделяет его всей полнотой власти над проектом и его финансированием.

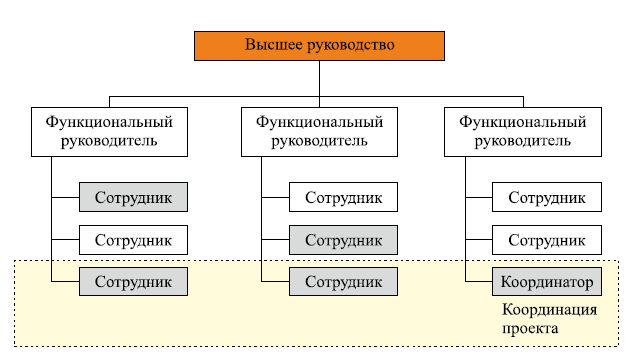


Рисунок 3.10. Слабая матричная структура

В слабой матрице роль и полномочия сотрудника, который координирует проект, сильно ограничены. Реальное руководство проектом осуществляет один из функциональных руководителей. Координатор проекта, его еще часто называют «трекер», помогает этому руководителю собирать информацию о статусе выполняемых проектных работ, учитывает затраты, составляет отчеты.

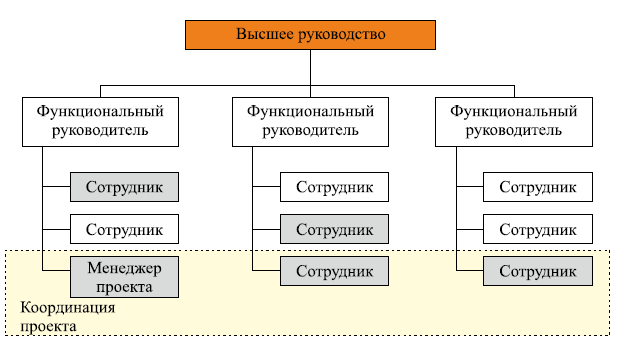


Рисунок 3.11. Сбалансированная матричная структура

Сбалансированная матрица характеризуется тем, что появляется менеджер проекта, который реально управляет выделенными на проект ресурсами. Он планирует работы, распределяет задачи среди исполнителей, контролирует сроки и результаты, несет полную ответственность за достижение целей проекта, при соблюдении ограничений.

В сбалансированных матрицах наиболее ярко проявляется проблема двойного подчинения. Руководитель функционального подразделения и менеджер проекта имеют примерно равное влияние на материальный и профессиональный рост разработчиков.

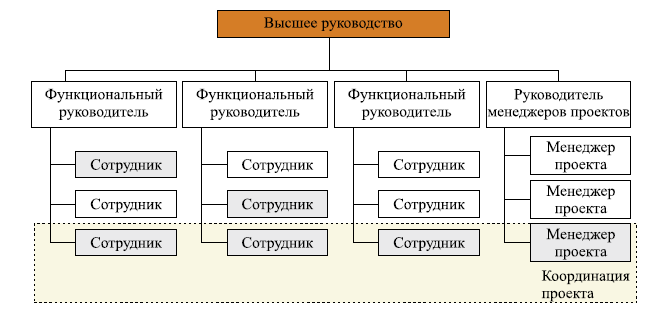


Рисунок 3.12. Сильная матричная структура

*В сильной матрице признается, что проектное управление является самостоятельной областью компетенции, в которой необходимо накапливать экспертизу и использовать общие ресурсы. Поэтому в сильной матрице менеджеры проектов объединяются в самостоятельное функциональное подразделение -* ***офис управления проектами (ОУП)*.** ОУП разрабатывает корпоративные политики и стандарты в области проектного управления, планирует и осуществляет профессиональное развитие менеджеров.

**Одной из особенностей матричных структур является то, что они становятся «плоскими», исчезает многоступенчатая иерархия**. Предприятие, как правило, делится на функциональные отделы, в которых работают специалисты разных категорий, напрямую подчиняющиеся начальнику отдела. Начальники лабораторий, секторов, групп упраздняются за ненадобностью.

В матричных структурах ***роль начальника функционального подразделения в производственном процессе заметно снижается***, по сравнению с функциональными структурами. В его компетенции остаются вопросы стратегического развития функционального направления, планирование и развитие карьеры сотрудников, вопросы материально-технического обеспечения работ. Следует учитывать, что такое перераспределение полномочий и ответственности от функциональных руководителей к менеджерам проектов часто служит источником конфликтов в компаниях при их переходе от функциональной структуры к матричной.

***В разработке ПО наиболее распространена матричная организация.*** Причем, в компаниях, которые занимаются продуктовой разработкой ПО, функциональные подразделения определяются в соответствие с линейкой продуктов. Например, отдел разработки CRM-систем, отдел разработки финансовых систем, отдел разработки дополнительных продуктов.

В компаниях, которые ориентированы в основном на заказную разработку ПО, функциональные подразделения чаще объединяются в соответствие с используемыми информационными технологиями. Например, отдел разработки баз данных, отдел разработки J2EE-приложений, отдел веб-разработок, отделы тестирования, документирования и т.д.

В ***проектной организации*** (рисунок 3.13) члены команды часто располагаются в одном месте, большинство ресурсов организации вовлечено в работы по проекту, а менеджеры проектов имеют большую долю независимости и полномочий.

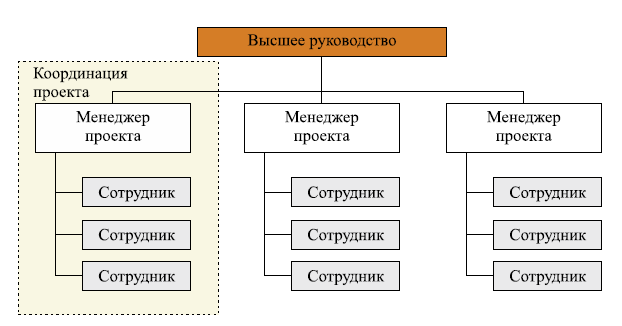


Рисунок 3.13. Проектная структура

В чисто проектных организациях:

 Проект организуется как самостоятельное производственное подразделение.

 Персонал на проект набирается по временным контрактам.

 После завершения проекта персонал увольняется.

 Медленный старт.

 Опыт не аккумулируется.

 Команды не сохраняются.

Проектные организации не самые эффективные, но порой *единственно возможные для выполнения проектов, которые физически удалѐны от исполняющей организации*, например, строительство нового нефтепровода.

Многие организации используют все эти структуры на различных уровнях, как показано на рисунке 3.14 (**комбинированная организация**). Например, даже фундаментально функциональная организация может создать специальную команду проекта для выполнения критически важного проекта.

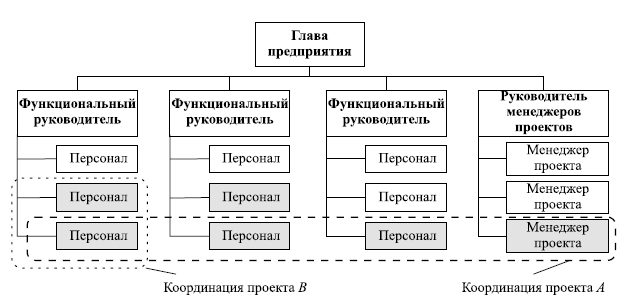


Рисунок 3.14. Комбинированная структура

Реализация проекта может потребовать выполнения установленного количества всевозможных мероприятий и работ, которые могут быть разделены на две группы:

● основная деятельность,

● деятельность по обеспечению проекта.

Данное разделение не может служить делением процесса осуществления проекта на фазы и стадии, так как предполагаемая деятельность зачастую совпадает во времени.

К основной деятельности обычно относят:

● анализ проблемы,

● формирование целей проекта,

● базовое и детальное проектирование,

● выполнение технических работ,

● ремонт,

● сдачу проекта,

● эксплуатацию проекта,

● обслуживание и демонтаж оборудования и т. п.

***Основная литература***

1. Архипенков С. Лекции по управлению программными проектами// Электронный ресурс:http://citforum.ru/SE/project/

2. Основы управления ИТ-проектами [Электронный ресурс]: учебное пособие / сост. Е.Р. Кирколуп, Ю.Г., Скурыдин, Е.М. Скурыдина. – Барнаул : АлтГПУ, 2017// Электронный ресурс:http://library.altspu.ru/dc/pdf/kirkolup.pdf

3. Рыбалова Е.А. Управление проектами: учебное пособие / Е.А. Рыбалова. - Томск: ФДО, ТУСУР, 2015. - 205 с.

4. Богданов В.В. Управление проектами в Microsoft Project 2007. Учебный курс. -  СПб: Питер, 2010

[5. Коптелов А., Никитин И., Цулая М. Гибкая процессная методология Agile](https://www.intuit.ru/studies/courses/3590/832/info). - М.: ИНТУИТ // Электронный ресурс: // www.intuit.ru/studies/courses/3590/832/print\_lecture/31170

6. Топ-7 методов управления проектами: Agile, Scrum, Kanban, PRINCE2 и другие// www.pmservices.ru/project-management-news/top-7-metodov-upravleniya-proektami-agile-scrum-kanban-prince2-i-drugie/

### 7.[Грекул](https://www.intuit.ru/intuituser/userpage/v_i_grekul) В., [Коровкина](https://www.intuit.ru/intuituser/userpage/n_l_korovkina) Н., [Куприянов](https://www.intuit.ru/intuituser/userpage/y_v_kupriyanov) Ю. [Методические основы управления ИТ-проектами](https://www.intuit.ru/studies/courses/646/502/info). - М.: ИНТУИТ // Электронный ресурс: https://www.intuit.ru/studies/courses/646/502/info

### ***Дополнительная литература:***

8. Сериков А. Основы управления проектами. - М.: ИНТУИТ. Электронный ресурс: www.intuit.ru/studies/professional\_retraining/965/courses/272/info

9. Управление проектами: Учебное пособие / М.В. Романова. - М.: ИД ФОРУМ: НИЦ ИНФРА-М, 2014

10. Руководство к Своду знаний по управлению проектами РМВОК (A Guide to the Project Management Body of Knowledge PMBOK®) — 4-е изд. — Project Management Institute, Inc., 2008. — 464 с.

11. https://products.office.com/

**Лекция 4. Организация проектной деятельности (классический подход)**

*План:*

1. Жизненный цикл проекта

2. Фазы программного проекта

3. Деятельность по обеспечению проекта

**1. Жизненный цикл проекта**

**Цикл проекта (ЦП)** выступает основным элементом представления проектного анализа.

**Жизненный цикл проекта** — это набор, как правило, последовательных и иногда перекрывающихся фаз проекта, названия и количество которых определяются потребностями в управлении и контроле организации или организаций, вовлеченных в проект, характером самого проекта и его прикладной областью.

**Жизненный цикл проекта** – это время от первой затраты до последней выгоды проекта. В целом жизненный цикл проекта показывает развитие работы, которая ведется на различных этапах подготовки, реализации и эксплуатации проекта. В описание ЦП входит определение различных этапов разработки и реализации проекта.

***ЦП является некой определенной схемой или алгоритмом, посредствам которого устанавливается определенная последовательность действий на этапах разработки и внедрения проекта.***

Существенным *при выделении фаз, стадий и этапов проекта может стать обозначение отдельных контрольных точек, при прохождении которых извлекается дополнительная (внешняя) информация и устанавливаются или оцениваются вероятные направления развития проектов.*

Жизненный цикл обеспечивает базовую структуру для управления проектом, независимо от включенных в него конкретных работ.

Согласно **классическому подходу к разработке,** независимо от размеров и степени сложности, все проекты могут быть представлены в виде **жизненного цикла проекта** (рисунок 4.1 а,б) со следующей структурой:

1. Инициация (начало проекта)
2. Планирование (организация и подготовка)
3. Реализация (выполнение работ)
4. Завершение проекта.

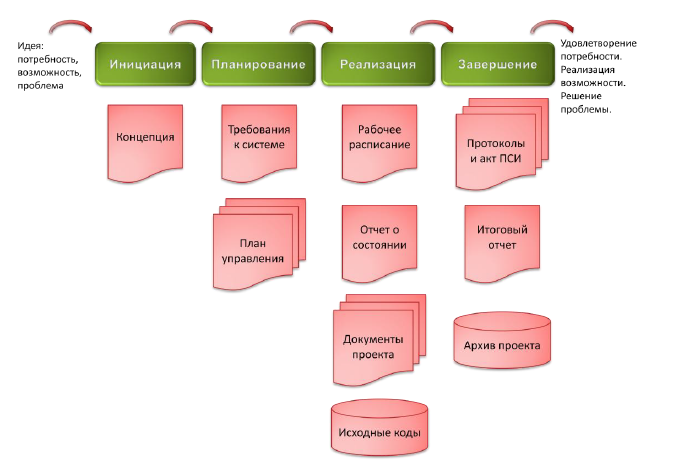


Рисунок 4.1 (а). Жизненный цикл и основные документы программного проекта

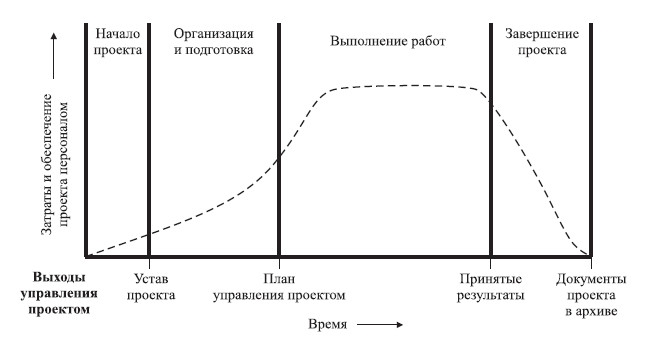


Рисунок 4.1, б. Типовые уровни затрат и ресурсов на протяжении жизненного цикла проекта

*На этапе инициации проекта* необходимо понять, что и зачем мы будем делать – разработать концепцию проекта.

*Этап планирования* определяет, как мы будем это делать.

*Этап реализации* - разработка, документация и тестирование программного продукта.

И, наконец, *на этапе завершения* мы должны подтвердить, что мы разработали именно тот продукт, который задумали в концепции проекта, а также провести приемо-сдаточные испытания (ПСИ) продукта на предмет соответствия его свойств, определенным ранее требованиям.

**2. Фазы программного проекта:**

1. Требования,
2. Проектирование,
3. Реализация,
4. Интеграция (Сборка),
5. Отладка,
6. Внедрение,
7. Сопровождение.

Жизненный цикл ***продукта*** обычно состоит из последовательных, неперекрывающихся фаз продукта, определяемых потребностью производства и контроля организации. Последней фазой жизненного цикла продукта, как правило, является прекращение сервисного обслуживания и поддержки. Обычно жизненный цикл проекта заключен в рамках жизненных циклов одного или нескольких продуктов. Но, тем не менее, *следует отличать жизненный цикл проекта от жизненного цикла продукта* (например, жизненные циклы программного проекта (ПП), информационного конечного продукта (ИКП), материального конечного продукта (МКП) и ЖЦ услуги.

**Фазы проекта** — это отдельные части в рамках проекта, требующие дополнительного контроля для эффективного управления достижением основного результата проекта.

Фазы проекта обычно выполняются последовательно, но в некоторых случаях могут перекрываться. Высокоуровневый характер фаз проекта превращает их в элемент жизненного цикла проекта. Фаза проекта не является группой процессов управления проектом.

Структура фаз позволяет разделить проект на логические подгруппы для более легкого управления, планирования и контроля. Независимо от количества фаз, составляющих проект, все фазы имеют схожие *характеристики*:

• При последовательном выполнении этапов завершение фазы (естественная точка) сопровождается определенного рода передачей полученного продукта в качестве результата фазы. Эти точки называются выходами фаз, контрольными событиями, воротами решений, точками критического анализа или остановки.

• Как правило, работы фазы имеют свойства, которые отличают ее от других фаз. При этом могут привлекаться разные организации и использоваться разные наборы навыков.

• Для успешного достижения главного результата или цели фазы требуется дополнительная степень контроля. Повторение процессов во всех пяти группах процессов обеспечивает такую дополнительную степень контроля и определяет границы фазы.

**Деятельность по обеспечению проекта,** в свою очередь, может быть разделена на:

● организационную,

● правовую,

● кадровую,

● финансовую,

● материально-техническую,

● коммерческую,

● информационную.

Отчетливого и однозначного распределения данных работ в логической последовательности и во времени не существует (это относится также к фазам и этапам выполнения проекта), так как решающими являются цели и условия реализации проекта.

**Программой промышленного развития ООН (UNIDO) предоставлено собственное видение *проекта как цикла***, *состоящего из трех отдельных фаз:* прединвестиционной, инвестиционной и эксплуатационной.

***Прединвестиционная фаза*** представлена следующими стадиями:

● определение инвестиционных возможностей, анализ альтернативных вариантов,

● предварительный выбор проекта – предварительное технико-экономическое обоснование, выводы по проекту и решение об инвестировании.

***Инвестиционная фаза*** представлена следующими стадиями:

● установление правовой, финансовой и организационной основ для осуществления проекта,

● приобретение и передача технологий,

● детальная проектная обработка и составление контрактов,

● приобретение земли,

● строительные работы и установка оборудования,

● предпроизводственный маркетинг,

● набор и обучение персонала,

● сдача в эксплуатацию и запуск.

***Фаза эксплуатации*** анализируется с точки зрения долгосрочных планов, а также и краткосрочных. В *краткосрочном плане* анализируется вероятное возникновение проблем, которые могут быть связаны с применением предпочтенной технологии, функционированием оборудования или с квалификацией персонала. В *долгосрочном плане* к анализу берется выбранная стратегия и определенные затраты на производство и маркетинг, а также полученная прибыль от продаж.

**Общим аспектом к распределению работ, принадлежащих к различным фазам и стадиям ЦП**, может стать ***подход Всемирного банка***. На рисунке 4.2 представлено шесть основных стадий, представляющих существенную роль в большей части проектов. Данные стадии обозначаются как: идентификация, разработка, экспертиза, переговоры, реализация и завершающая оценка. Настоящие стадии представлены двумя фазами:

● фаза проектирования – идентификация, разработка, экспертиза;

● фаза внедрения – переговоры, реализация и завершающая оценка.

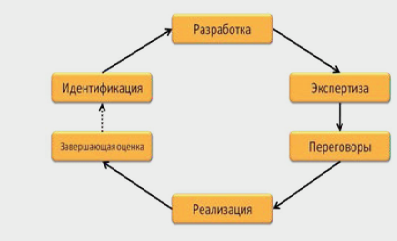
**

Рисунок 4.2. Цикл проекта

**Идентификация –** представляет собой первую стадию цикла проекта. Идентификация – имеет отношение к выбору или генерировании подобных значительных идей, позволяющие обеспечить выполнение существенных задач развития. Концепция проекта может быть определена:

● деятельностью частных или государственных организаций, желающих приобрести преимущества в применении новейших возможностей;

● затруднениями или удерживаниями в ходе разработки, которые порождены нехваткой существенных производственных мощностей, малоразвитостью сервиса, нехваткой материальных и человеческих ресурсов или же административными или другими препятствиями;

● готовностью сформировать подходящие условия для создания соответственной инфраструктуры производства и управления;

● готовностью реализовать задачи, поставленные перед предприятием;

● потребностями и поиском допустимых путей их реализации;

● природными катаклизмами (наводнения, засухи, ураганы и землетрясения);

● возможностью предоставления полностью или частично неиспользованных материальных или человеческих ресурсов и вероятностью их применения в других областях;

● потребностью произвести дополнительные денежные вложения.

*Проект может считаться безошибочным и может быть передан на стадию разработки*, в случае, если выполнены следующие условия:

● осуществлен отбор различных вариантов проекта;

● определены существенные проблемы, которые могут оказать влияние на участь проекта, и установлено, что они могут быть разрешимы;

● установлены возможные выгоды и затраты;

● определено существование совместной поддержки руководства и прочих участников проекта.

**2.** Дальнейшей стадией цикла проекта является **разработка *- д****етализация целей проекта и тех средств, с помощью которых эти цели достигаются, представляет существенную часть работы по созданию проекта.* Существуют различные пути достижения цели на данной стадии цикла. По мере детализации целей проекта и уменьшения количества вариантов и вариантов их осуществления проект становится конкретизированным, и его создание ведется на базе предпочтительных вариантов.

В процессе создания проекта определенное место занимает *скрининг* – основной анализ осуществления проекта. Полагая, что необходимость подобного анализа очевидная, когда речь идет о значительных инфраструктурных проектах. Подобная проверка позволяет определить, есть ли необходимость для осуществления проекта и который из вариантов проекта является оптимальным для достижения его целей.

**3. Экспертизу** проекта должны осуществлять не только люди, проводящие анализ и разработку проекта, но и сторонние эксперты. Экспертиза дает возможность провести доскональный анализ всех сторон проекта и его результатов. На данном этапе закладывается ядро для осуществления проекта. План проекта, установленный на этапе экспертизы, является основой для оценивания благополучности проекта. Экспертизе могут подвергаться и сам проект, и организация, реализовавшая данный проект. Задачей аудита проекта является установление того, насколько основательные результаты проекта превзойдут его негативные последствия.

**4.** После стадии экспертизы проекта представители инвесторов проводят официальную встречу, т.е. **переговоры**, чтобы подтвердить сроки и условия его финансирования. Эти договоренности потом формулируются как правовые обязательства, изложенные в документах, которые подписываются обеими сторонами.

**5. Реализация** проекта начинается с планирования. В плане осуществления проекта существенной частью является система достижения согласия относительно распределения ролей, ответственности и прав всех участников проекта.

Следующим этапом реализации являются ***проведение переговоров и составление договоров*** на поставку сырья и технологий, оборудования, материалов, а также составление соглашений на выполнение субконтрактных работ. На этой стадии может быть реализовано ***инженерно-техническое проектирование*** (дизайнерские работы), строительство как самого объекта, так и необходимых инфраструктурных элементов проекта, производственный маркетинг и обучение.

Также существенным на стадии реализации является **контроль**. Выделяются три аспекта проверки проекта. Во-первых, инженерно-технический надзор за техническими аспектами проекта, который выполняют После стадии реализации, необходимо оценить риски проекта и вероятный вклад проекта в стабильность жизни людей.

**6.** **Завершающая оценка** предусматривает ретроспективный анализ проекта. Она ведется преимущественно тогда, когда проект после реализации находился в эксплуатации от двух до трех лет.

***Основная литература***

1. Архипенков С. Лекции по управлению программными проектами// Электронный ресурс:http://citforum.ru/SE/project/

2. Основы управления ИТ-проектами [Электронный ресурс]: учебное пособие / сост. Е.Р. Кирколуп, Ю.Г., Скурыдин, Е.М. Скурыдина. – Барнаул : АлтГПУ, 2017// Электронный ресурс:http://library.altspu.ru/dc/pdf/kirkolup.pdf

3. Рыбалова Е.А. Управление проектами: учебное пособие / Е.А. Рыбалова. - Томск: ФДО, ТУСУР, 2015. - 205 с.

4. Бухарев Н.Р. Введение в гибкое управление проектами. Краткий конспект лекций// tulpar.kfu.ru/course/view.php?id=2284

[5. Коптелов А., Никитин И., Цулая М. Гибкая процессная методология Agile](https://www.intuit.ru/studies/courses/3590/832/info). - М.: ИНТУИТ // Электронный ресурс: // www.intuit.ru/studies/courses/3590/832/print\_lecture/31170

6. Топ-7 методов управления проектами: Agile, Scrum, Kanban, PRINCE2 и другие// www.pmservices.ru/project-management-news/top-7-metodov-upravleniya-proektami-agile-scrum-kanban-prince2-i-drugie/

### 7.[Грекул](https://www.intuit.ru/intuituser/userpage/v_i_grekul) В., [Коровкина](https://www.intuit.ru/intuituser/userpage/n_l_korovkina) Н., [Куприянов](https://www.intuit.ru/intuituser/userpage/y_v_kupriyanov) Ю. [Методические основы управления ИТ-проектами](https://www.intuit.ru/studies/courses/646/502/info). - М.: ИНТУИТ // Электронный ресурс: https://www.intuit.ru/studies/courses/646/502/info

### ***Дополнительная литература:***

8. Сериков А. Основы управления проектами. - М.: ИНТУИТ. Электронный ресурс: www.intuit.ru/studies/professional\_retraining/965/courses/272/info

9. Управление проектами: Учебное пособие / М.В. Романова. - М.: ИД ФОРУМ: НИЦ ИНФРА-М, 2014

10. Руководство к Своду знаний по управлению проектами РМВОК (A Guide to the Project Management Body of Knowledge PMBOK®) — 4-е изд. — Project Management Institute, Inc., 2008. — 464 с.

11. https://products.office.com/

**Лекция 5. Инициация проекта**

*План:*

1. Управление приоритетами проектов

1.1. Финансовая ценность

1.2. Стратегическая ценность

1.3. Уровень рисков

2. Концепция проекта

2.1.Цели и результаты проекта

2.2 Допущения и ограничения

2.3 Ключевые участники и заинтересованные стороны

2.4 Ресурсы

2.5 Сроки

2.6 Риски

2.7 Критерии приемки

2.8. Обоснование полезности проекта

3. Процессы управления проектом

**1. Управление приоритетами проектов**

Эффективные процессы инициации программного проекта минимум наполовину определяют его будущую успешность. Недостаточное внимание именно этой фазе проекта неизбежно приводит к существенным проблемам при планировании, реализации и завершении проекта.

***Инициация состоит из процессов, способствующих формальной авторизации начала нового проекта или фазы проекта.*** Процессы инициации часто выполняются вне рамок проекта и связаны с организационными, программными или портфельными процессами.

*В ходе процесса инициации уточняются первоначальное описание содержания и ресурсы, которые организация планирует вложить*. На этом этапе также выбирается *менеджер проекта*, если он еще не назначен, и документируются исходные допущения и ограничения. Эта информация заносится в *Устав проекта* и, если он одобряется, проект официально авторизуется.

***Устав проекта*** - документ, выпущенный инициатором или спонсором проекта, который формально узаконивает существование проекта и предоставляет менеджеру проекта полномочия использовать организационные ресурсы в операциях проекта.

В российской практике данный документ чаще называется **Концепция проекта.** *Концепция* (от лат. conceptio — понимание, система), определённый способ понимания, трактовки какого-либо предмета, явления, процесса, основная точка зрения на предмет и др., руководящая идея для их систематического освещения.

В компании, которая принимает решение о старте того или иного проекта разработки ПО, должна существовать единая система критериев для оценки его значимости.

Система критериев оценки проекта должна позволять из множества возможных для реализации проектов выбрать наиболее приоритетные для компании.

**Приоритет любого проекта** должен определяться **на основе оценки трех его характеристик**:

1. Финансовая ценность.
2. Стратегическая ценность.
3. Уровень рисков.

**1.1. Финансовая ценность**

Шкала оценки финансовой ценности проекта может выглядеть следующим образом:

* *Высокая*. Ожидаемая окупаемость до 1 года. Ожидаемые доходы от проекта не менее чем в 1.5 раз превышают расходы. Все допущения при проведении этих оценок четко обоснованы.
* *Выше среднего*. Ожидаемая окупаемость проекта от 1 года до 3 лет. Ожидаемые доходы от проекта не менее чем в 1.3 раза превышают расходы. Большинство допущений при проведении этих оценок имеют под собой определенные основания.
* *Средняя*. Проект позволяет улучшить эффективность производства в Компании и потенциально может снизить расходы компании не менее чем на 30%. Проект может иметь информационную ценность или помочь лучше контролировать бизнес.
* *Низкая*. Проект немного снижает расходы компании не менее чем на 10% и дает некоторые улучшения производительности производства.

Например. Финансовая ценность проектов разработки ПО, проектов внедрения или сопровождения, которые выполняются в соответствие с заключенными коммерческими договорами, может быть оценена как высокая. Проект планового развития функциональности продуктов в соответствии с требованиями рынка, инициируемое менеджером продукта на основе анализа предложений отделов маркетинга, консалтинга, продаж и технической поддержки, может получить оценку финансовой ценности выше среднего, а проекты изменения технологических процессов или проекты внутренней автоматизации могут иметь среднюю финансовую ценность.

Одной финансовой ценности для определения приоритета проекта недостаточно. Например, ни одна компания разработчик ПО не возьмется за автоматизацию нелегального оборота наркотиков, если это не соответствует стратегии ее бизнеса. Поэтому, важным показателем приоритета проекта является его соответствие стратегическим целям компании.

**1.2. Стратегическая ценность**

Шкала оценки стратегической ценности проекта может иметь следующий вид:

* *Высокая*. Обеспечивает стратегическое преимущество, дает устойчивое увеличение рынка или позволяет выйти на новый рынок. Решает значительные проблемы, общие для большинства важных клиентов. Повторение конкурентами затруднено или потребует от 1 до 2 лет.
* *Выше среднего.* Создает временные конкурентные преимущества. Выполнение обязательств перед многими важными клиентами. Конкурентное преимущество может быть удержано в течение 1 года.
* *Средняя.* Поддерживается доверие рынка к компании. Повышает мнение клиентов о качестве предоставляемых услуг или способствует выполнению обязательств перед несколькими клиентами. Конкуренты уже имеют или способны повторить новые возможности в пределах года.
* *Низкая.* Стратегическое воздействие отсутствует или незначительно. Влияние на клиентов несущественно. Конкуренты могут легко повторить результаты проекта.

**1.3. Уровень рисков**

Третьим обязательным показателем приоритета проекта должна быть **оценка уровня его риска**. Ни один проект, который имеет даже самую высокую оценку финансовой выгодности, не будет запущен в производство, если достижение этой сверхвыгоды имеет минимальные шансы.

Примерная шкала оценки уровня рисков проекта может иметь следующий вид:

* *Низкий*. Цели проекта и требования хорошо поняты и документированы. Масштаб и рамки проекта заданы четко. Ресурсы требуемой квалификации доступны в полном объеме. Разрабатываемые системы не потребуют новой технологической платформы.
* *Средний.* Цели проекта определены более-менее четко. Хорошее понимание требований к системе. Масштаб и рамки проекта заданы достаточно хорошо. Ресурсы требуемой квалификации доступны в основном. Системы создаются на новой, но стабильной технологической платформе.
* *Выше среднего.* Цели проекта недостаточно четки. Задачи системы или бизнес-приложения поняты недостаточно полно. Понимание масштаба и рамок проекта недостаточно. Ресурсы требуемой квалификации сильно ограничены. Системы создаются на новой технологической платформе, сомнения в рыночной стабильности платформы.
* *Высокий.* Цели проекта нечетки. Основные функциональные компоненты системы не определены. Масштаб и рамки проекта непонятны. Ресурсы требуемой квалификации практически отсутствуют. Системы создаются на новой технологической платформе, в отношении которой крайне мало ясности. Технологии имеют неподтвержденную стабильность.

Если компания уделяет мало внимания управлению приоритетами своих проектов, то это приводит к переизбытку реализуемых проектов, перегруженности исполнителей, постоянным авралам и сверхурочным работам и, как следствие, к низкой эффективности производственной деятельности. При старте нового проекта с высоким приоритетом, компания должна остановить или закрыть менее значимые проекты, чтобы обеспечить новый проект необходимыми ресурсами, а не пытаться сделать все и сразу за счет интенсификации работ, как правило, это не получается.

**2. Концепция проекта**

У каждого проекта должна быть концепция. Если проект небольшой, то для изложения концепции часто достаточно несколько абзацев. Однако, стартовать проект без концепции, это все равно, что отправлять корабль в плавание, не определив для него пункт назначения.

Концепция проекта разрабатывается на основе анализа потребностей бизнеса. Главная функция документа - подтверждение и согласование единого видения целей, задач и результатов всеми участниками проекта. Концепция определяет ***что*** *и* ***зачем***делается в проекте.

Концепция проекта это ключевой документ, который используется для принятия решений в ходе всего проекта, а также на фазе приемки - для подтверждения результата. Она содержит, как правило, следующие разделы:

* Название проекта
* Цели проекта
* Результаты проекта
* Допущения и ограничения
* Ключевые участники и заинтересованные стороны
* Ресурсы проекта
* Сроки
* Риски
* Критерии приемки
* Обоснование полезности проекта

***Пример***

В качестве примера, который позволит иллюстрировать теоретическое изложение основ управления проектами, возьмем реальный проект разработки ПО для автоматизации одного из подразделений крупной производственной компании. Назовем его «Автоматизированная система продажи документации».

Краткая легенда проекта. Заказчик ОАО «XYZ» является одним из ведущих производителей сложных технических изделий. Отдел «123», входящий в ОАО «XYZ», отвечает за продажу дополнительной сопроводительной документации для клиентов ОАО.

Дополнительная документация не входит в стандартную поставку, поскольку владелец этого технического изделия не всегда сам его эксплуатирует, а передает в эксплуатацию другой компании, которая становится клиентом «XYZ», и закупает у нее эксплуатационную документацию. Ремонт и техобслуживание конкретного изделия может выполнять третья компания, которой уже потребуется детальная техническая документация по ремонту и обслуживанию. Она также становится клиентом «XYZ» и закупает у нее требуемую продукцию.

Основная функция отдела «123» — получение и обработка заказов на дополнительную документацию, согласно ежегодно рассылаемому каталогу. В связи с переездом отдела «123» в новое здание, была поставлена задача на разработку и поставку системы, автоматизирующей основную деятельность отдела «123».

Текст документа Концепция проекта, который будет приводиться в качестве примера, будем выделять цветом.

**2.1.Цели и результаты проекта**

Цели проекта должны отвечать на вопрос, ***зачем***данный проект нужен. Цели проекта должны описывать бизнес-потребности и задачи, которые решаются в результате исполнения проекта. Целями проекта могут быть:

* Изменения в Компании. Например, автоматизация ряда бизнес-процессов для повышения эффективности основной производственной деятельности
* Реализация стратегических планов. Например, завоевание значительной доли растущего рынка за счет вывода на него нового продукта.
* Выполнение контрактов. Например, разработка программного обеспечения по заказу.
* Разрешение специфических проблем. Например, доработка программного продукта в целях приведения его в соответствие с изменениями в законодательстве.

*Цели должны быть значимыми* (направленными на достижение стратегических целей Компании), конкретными (специфичными для данного проекта), измеримыми (т.е иметь проверяемые количественные оценки), реальными (достижимыми). Четкое определение бизнес-целей важно, поскольку существенно влияет на все процессы и решения в проекте. Проект должен быть закрыт, если признается, что достижение цели невозможно или стало нецелесообразным. Например, если реальные затраты на проект будут превосходить будущие доходы от его реализации.

***Результаты проекта*** отвечают на вопрос, ***что***должно быть получено после его завершения. Результаты проекта должны определять:

* Какие именно бизнес-выгоды получит заказчик в результате проекта.
* Какой продукт или услуга. Что конкретно будет произведено по окончании проекта.
* Высокоуровневые требования. Краткое описание и при необходимости ключевые свойства и/или характеристики продукта/услуги.

Следует помнить, что результаты проекта должны быть измеримыми. Это означает, что при оценке результатов проекта должна иметься возможность сделать заключение достигнуты оговоренные в концепции результаты или нет.

***Пример.***Соответствующий раздел документа концепция проекта создания «Автоматизированной системы продажи документации» будет выглядеть следующим образом.

1. Цели и результаты проекта

1.1. Целью проекта является повышение эффективности основной производственной деятельности отдела «123».   
1.2. Дополнительными целями проекта являются:   
1.2.1. Установление долгосрочных отношений с важным заказчиком ОАО «XYZ».   
1.2.2. Выход на новый перспективный рынок современных B2C систем.

1. Результаты проекта должны обеспечить:

2.1. Снижение затрат на обработку заявок.   
2.2. Снижение сроков обработки заявок.   
2.3. Повышение оперативности доступа к информации о наличии продукции.   
2.4. Повышение оперативности доступа к информации о прохождении заявок.   
2.5. Повышение надежности и полноты хранения информации о поступивших заявках и результатах их обработки.

1. Продуктами проекта являются:

3.1. Прикладное ПО и документация пользователей.   
3.2. Базовое ПО.   
3.3. Оборудование ЛВС, рабочие станции, сервера и операционно-системное ПО.   
3.4. Проведение пуско-наладочных работ и ввод в опытную эксплуатацию.   
3.5. Обучение пользователей и администраторов системы.   
3.6. Сопровождение системы на этапе опытной эксплуатации.   
3.7. Передача системы в промышленную эксплуатацию.

1. Система должна автоматизировать следующие функции:

4.1. Авторизация и аутентификация пользователей.   
4.2. Просмотр каталога продуктов.   
4.3. Поиск продуктов по каталогу.   
4.4. Заказ выбранных продуктов.   
4.5. Просмотр информации о статусе заказа.   
4.6. Информирование клиента об изменении статуса заказа.   
4.7. Просмотр и обработка заказов исполнителями из службы продаж.   
4.8. Просмотр статистики поступления и обработки заказов за период.   
4.9. Подготовка и сопровождение каталога продукции.

**2.2 Допущения и ограничения**

Данный раздел описывает исходные допущения и ограничения. Допущения, как правило, тесно связаны с управлением рисками, о котором мы будем говорить далее. В разработке ПО часто приходится формулировать риски в виде допущений, тем самым передавая его заказчику. Например, оценивая проект разработки и внедрения по схеме с фиксированной ценой, мы должны записать в допущения предположение о том, что стоимость лицензий на стороннее ПО не изменится, до завершения проекта.

Ограничения, как правило, сокращают возможности проектной команды в выборе решений. В частности они могут содержать:

* Специфические нормативные требования. Например, обязательная сертификация продукта, услуги на соответствие определенным стандартам.
* Специфические технические требования. Например, разработка под заданную программно-аппаратную платформу.
* Специфические требования к защите информации.

В этом разделе также уместно сформулировать те требования к системе, которые могут ожидаться заказчиком по умолчанию, но не включаются в рамки данного проекта. Например, в данный раздел может быть включен пункт о том, что разработка программного интерфейса (API) для будущей интеграции с другими системами заказчика не входит в задачи данного проекта.

***Пример.***Содержание этого раздела для нашего проекта-примера выглядит следующим образом.

1. Допущения и ограничения

5.1. Проектирование прикладного ПО выполняется с использованием UML1.   
5.2. Средством разработки ПО является Symantec Visual Cafe for Java2.   
5.3. В качестве промежуточного ПО сопровождения и поддержки каталога используется ОО БД «Poet»3.   
5.4. Нагрузка на систему не должна быть более 100 одновременно работающих пользователей.   
5.5. В рамки проекта не входят:

5.5.1. Защита системы от преднамеренного взлома.   
5.5.2. Разработка B2B API и интеграция с другими системами.

[1](http://citforum.ru/SE/project/arkhipenkov_lectures/6.shtml" \l "footnote1_back) Проект стартовал в 2000 году, тема UML тогда была на слуху и даже оставались те, кто верил, что из модели на UML можно будет генерировать исходный код.

[2](http://citforum.ru/SE/project/arkhipenkov_lectures/6.shtml" \l "footnote2_back) Таково было требование Заказчика, поскольку этот инструмент использовали его программисты, которым предполагалось передавать систему на сопровождение.

[3](http://citforum.ru/SE/project/arkhipenkov_lectures/6.shtml" \l "footnote3_back) Еще один пример горячей темы и не оправдавшихся надежд — это объектно-ориентированные базы данных. У заказчика проекта уже были закуплены лицензии на эту базу данных и он очень хотел получить возврат от этих инвестиций. Поэтому ее использование в проекте стало одним из требований. К счастью, нам удалось быть достаточно убедительными и обосновать необходимость дополнительно использовать RDBMS Oracle для решения транзакционных задач.

**2.3 Ключевые участники и заинтересованные стороны**

Одна из задач фазы инициации проекта это выявить и описать всех его участников. К участникам проекта относятся все заинтересованные стороны (stakeholders), лица и организации, например заказчики, спонсоры, исполняющая организация, которые активно участвуют в проекте или чьи интересы могут быть затронуты при исполнении или завершении проекта. Участники также могут влиять на проект и его результаты поставки.

К ключевым участникам программного проекта, как правило, относятся:

*Спонсор проекта* - лицо или группа лиц, предоставляющая финансовые ресурсы для проекта в любом виде.

*Заказчик проекта* - лицо или организация, которые будут использовать продукт, услугу или результат проекта. Следует учитывать, что заказчик и спонсор проекта не всегда совпадают.

*Пользователи* результатов проекта.

*Куратор проекта* - представитель исполнителя, уполномоченный принимать решение о выделении ресурсов и изменениях в проекте.

*Руководитель проекта* - представитель исполнителя, ответственный за реализацию проекта в срок, в пределах бюджета и с заданным качеством.

*Соисполнители проекта*. Субподрядчики и поставщики.

***Пример.***Содержание этого раздела в концепции-примере будет иметь вид.

1. Ключевые участники и заинтересованные стороны

6.1. Спонсор проекта — директор Департамента информатизации ОАО «XYZ» В.Васильев.   
6.2. Заказчик — начальник Отдела «123» Ф.Федотов   
6.3. Пользователи автоматизированной системы:   
6.4. Клиенты ОАО «XYZ» (поиск и заказ документации).   
6.5. Руководство ОАО «XYZ» (анализ деятельности Отдела «123»).   
6.6. Сотрудники производственных департаментов ОАО «XYZ» (сопровождение каталога).   
6.7. Сотрудники Отдела «123» (обработка заявок и поставка документации).   
6.8. Сотрудники департамента информатизации ОАО «XYZ» (администрирование системы).   
6.9. Куратор проекта — начальник отдела заказных разработок И.Иванов.   
6.10. Руководитель проекта — ведущий специалист отдела заказных разработок МП П.Петров.

1. Соисполнители:

7.1. Поставщик оборудования и операционно-системного ПО — ООО «Альфа».   
7.2. Поставщик базового ПО — ООО «Бета».

**2.4 Ресурсы**

Для того чтобы понять, сколько будет стоить реализация программного проекта, требуется определить и оценить ресурсы необходимые для его выполнения:

* Людские ресурсы и требования к квалификации персонала.
* Оборудование, услуги, расходные материалы, лицензии на ПО, критические компьютерные ресурсы.
* Бюджет проекта. План расходов и, при необходимости, предполагаемых доходов проекта с разбивкой по статьям и фазам/этапам проекта.

Специфика программного проекта заключается в том, что людские ресурсы вносят основной вклад в его стоимость. Все остальные затраты, как правило, незначительны, по сравнению с этим расходами.

На фазе инициации хорошей считается оценка трудозатрат с точностью от -50% до +100%.

Необходимо помнить, что помимо непосредственно программирования в проекте разработки ПО есть много других процессов, которые требуют ресурсы соответствующей квалификации, а само программирование составляет лишь четверть всех затрат. Распределение трудозатрат по основным производственным процессам при современном процессе разработки ПО в среднем выглядит как на рисунке 5.1.

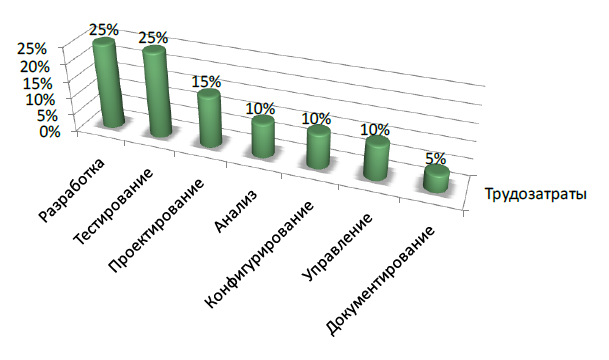


Рисунок 5.1. Распределение трудозатрат по основным производственным процессам при разработке ПО

Поэтому, если по вашей оценки для реализации требуемой функциональности в проекте необходимо написать 10 KSLOC (тысяч строк исходного программного кода), а ваши программисты пишут в среднем по 100 SLOC в день, то общие трудозатраты на проект будут не 100 чел.\*дней, а не менее чем 400 чел.\*дней. Остальные ресурсы потребуются на анализ и уточнение требований, проектирование, документирование, тестирование и другие проектные работы.

Прежде, чем определять численность и состав проектной команды для нашего примера, нам необходимо сделать оценку трудоемкости разработки ПО.

***Пример.***В нашем случае такая экспертная оценка составила с учетом затрат на гарантийное сопровождение на этапе опытной эксплуатации 9000 чел.\*час. Исходя из эмпирической кривой Б. Боэма (Рисунок 5.2), численность команды, близкая к оптимальной, составила 10 человек, из них

1. Ресурсы проекта

8.1. Требования к персоналу

8.1.1.  1 — руководитель проекта,   
8.1.2.  1 — технический лидер (архитектура, проектирование),   
8.1.3.  1 — системный аналитик (требования, тест-дизайн, документирование),   
8.1.4.  4 — программисты (с учетом работ по конфигурационному управлению),   
8.1.5.  3 — тестировщики.

8.2. Материальные и другие ресурсы

8.2.1. Сервер управления конфигурациями и поддержки системы контроля версий   
8.2.2. 2 серверных комплекса (для разработки и тестирования):   
8.2.3. Сервер приложений с установленным BEA Weblogic AS   
8.2.4. Сервер оперативной БД с установленной Oracle RDBMS   
8.2.5. Сервер каталога с установленной OODB "Poet"

8.3. Лицензии на средства разработки и тестирования:

8.3.1. Oracle Designer — 1 лицензия   
8.3.2. Symantec Visual Cafe for Java — 5 лицензий.   
8.3.3. IBM Rational Test Robot (1 лицензия разработчика + неограниченная лицензия на клиент).

8.4. Расходная часть бюджета проекта

8.4.1. Разработка и сопровождение прикладного ПО:

8.4.1.1. 9000 чел.\*час. \* $40 = $360 000

8.4.2. Поставка оборудования и операционно-системного ПО:

8.4.2.1. 3 сервера \* $10 000 = $30 000

8.4.3. Поставка базового ПО:

8.4.3.1. BEA Weblogic AS $20 000   
8.4.3.2. Oracle RDBMS $20 000

Итого: $430 000

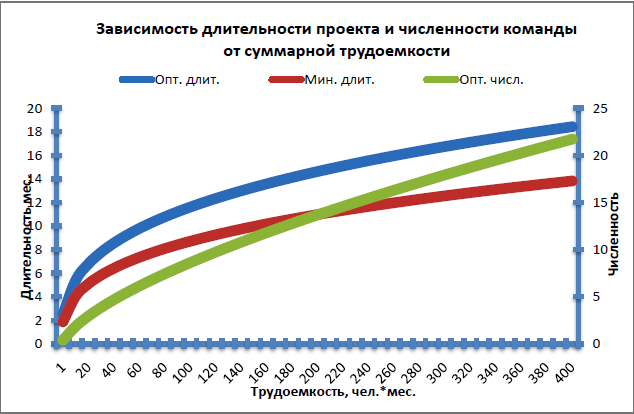


Рисунок 5.2

**2.5 Сроки**

Ф. Брукс  писал: «Чтобы родить ребенка требуется девять месяцев независимо от того, сколько женщин привлечено к решению данной задачи. Многие задачи программирования относятся к этому типу, поскольку отладка по своей сути носит последовательный характер».

Там же Брукс приводит исключительно полезную, но почему-то редко применяемую, эмпирическую формулу оценки срока проекта по его трудоемкости. Формула была выведена Барии Боэмом (BarryBoehm) на основе анализа результатов 63 проектов разработки ПО, в основном в аэрокосмической области. Согласно этой формуле, для проекта, общая трудоемкость которого составляет *N* ч.\*м. (человеко-месяцев), можно утверждать что:

* Существует оптимальное, с точки зрения затрат, время выполнения графика для первой поставки: T = 2,5 (N ч.\*м.)1/3. То есть оптимальное время в месяцах пропорционально кубическому корню предполагаемого объема работ в человеко-месяцах. Следствием является кривая, дающая оптимальную численность проектной команды (Рисунок 3).
* Кривая стоимости медленно растет, если запланированный график длиннее оптимального. Работа занимает все отведенное для нее время.
* Кривая стоимости резко растет, если запланированный график короче оптимального. Практически ни один проект невозможно завершить быстрее, чем за 3/4 расчетного оптимального графика вне зависимости от количества занятых в нем! (Рисунок 4)

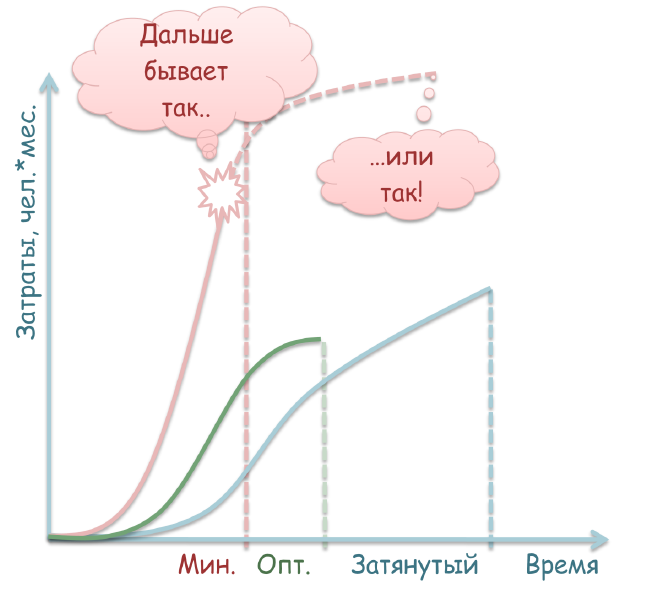


Рисунок 5.3. Следствия закона Б.Боэма

Этот примечательный результат дает менеджеру программного проекта солидное подкрепление, когда высшее руководство требует принятия невозможного графика.

Для серьезного программного проекта недостаточно определить только срок его завершения. Необходимо еще *определить его этапы – контрольные точки*, в которых будет происходить переоценка проекта на основе реально достигнутых показателей.

***Контрольная точка***- важный момент или событие в расписании проекта, отмечающее достижение заданного результата и/или начало / завершение определенного объема работы. Каждая контрольная точка характеризуется датой и объективными критериями ее достижения.

**Современный проект разработки ПО должен реализовываться с применением инкрементального процесса.** В этом случае контрольные точки должны соответствовать выпуску каждой промежуточной версии ПО, в которой будет реализована и протестирована определенная часть конечной функциональности программного продукта. В зависимости от сложности и масштаба проекта продолжительность одной итерации может составлять от 2 до 8 недель.

***Пример.***Соответствующий раздел концепции нашего проекта-примера будет иметь следующий вид.

1. Сроки проекта

9.1. 03.03 старт   
9.2. 28.11 завершение   
9.3. Контрольные точки:

9.3.1. 15.04 ТЗ утверждено   
9.3.2. 30.04 1-я итерация завершена. Подсистема заказа документации передана в тестовую эксплуатацию (на серверах разработчика).   
9.3.3. 15.05 Монтаж оборудования у заказчика завершен .   
9.3.4. 30.05 Базовое ПО установлено у заказчика.   
9.3.5. 15.06 2-я итерация завершена. Подсистема обработки заказов передана в тестовую эксплуатацию на оборудовании Заказчика   
9.3.6. 02.09 3-я итерация завершена. Акт передачи системы в опытную эксплуатацию утвержден   
9.3.7. 28.11 Система передана в промышленную эксплуатацию.

**2.6 Риски**

*Риск* - неопределенное событие или условие, наступление которого отрицательно или положительно сказывается на целях проекта.

Как правило, в случае возникновения негативного риска, почти всегда стоимость проекта увеличивается и происходит задержка в выполнении мероприятий, предусмотренных расписанием проекта.

На этапе инициации, когда нет необходимых данных для проведения детального анализа, часто приходится ограничиваться качественной оценкой общего уровня рисков: низкий, средний, высокий.

***Пример.***В случае нашего проекта-примера раздел «риски» будет выглядеть следующим образом.

1. Риски проекта

10.1. Задачи системы поняты недостаточно полно. Понимание масштаба и рамок проекта недостаточно. Системы создаются на новой технологической платформе, сомнения в рыночной стабильности платформы. Суммарный уровень рисков следует оценить выше среднего.

**2.7 Критерии приемки**

Критерии приемки должны определять числовые значения характеристик системы, которые должны быть продемонстрированы по результатам приемо-сдаточных испытаний или опытной эксплуатации и однозначно свидетельствовать о достижении целей проекта.

***Пример.***В рассматриваемом примере раздел «Критерии приемки» будет выглядеть следующим образом:

1. Критерии приемки. По итогам опытной эксплуатации система должна продемонстрировать следующие показатели:

11.1. Средние затраты сотрудников Отдела «123» на регламентную обработку одного заказа не превышают 4 чел.\*час.   
11.2. Срок регламентной обработки 1-го заказа не более 2 недель.   
11.3. Время поиска и предоставления информации о наличии дополнительной документации не более 1 мин.   
11.4. Время предоставления информации о сделанных заказах и истории их обработки не более 1 мин.   
11.5. Система хранит всю информацию о сделанных заказах и истории их обработки.   
11.6. Показатель доступности системы 98%.

**2.8. Обоснование полезности проекта**

Этот раздел концепции должен содержать краткое технико-экономическое обоснование проекта:

* Для кого предназначены результаты проекта.
* Описание текущей ситуации «As Is». Какие у потенциального заказчика существуют проблемы.
* Каким образом результаты проекта решают эти проблемы («To Be»).
* Насколько значимо для клиента решение данных проблем (оценка экономического эффекта).
* Какие преимущества в итоге из этого может извлечь компания-исполнитель проекта.

***Пример.***Соответствующий раздел в концепции проекта-примера будет иметь следующий вид.

1. Обоснование полезности проекта

12.1. Для Заказчика:

12.1.1. Повышение производительности обработки заказов в 2 раза.

12.1.1.1. "As Is": 2500 заказов/год по 8 чел.\*час.   
12.1.1.2. "To Be": 2500 заказов/год по 4 чел.\*час.   
12.1.1.3. Экономия: 2500 \* 4 \* $50 = $500 000 в год.

12.1.2. Повышение оперативности контроля

12.1.2.1. "As Is": Ежемесячная отчетность.   
12.1.2.2. "To Be": Отчетность on-line.

12.1.3. Повышение удовлетворенности клиентов:

12.1.3.1. Сокращение срока обработки заказа в 2 раза.   
12.1.3.2. Сокращение времени на поиск необходимой документации в 10 раз   
12.1.3.3. Повышение оперативности обновления каталога 10 раз.

12.2. Для компании-исполнителя:

12.2.1. Высокая стратегическая ценность. Дает устойчивое увеличение рынка и завоевание нового рынка.   
12.2.2. Финансовая ценность выше среднего. Ожидаемые доходы от проекта не менее чем в 1.3 раза превышают расходы.

Таким образом, эффективные процессы инициации программного проекта во многом определяют его будущую успешность. Недостаточное внимание этой фазе проекта неизбежно приводит к существенным проблемам при планировании, реализации и завершении.

Концепция проекта это ключевой документ, который используется для принятия решений в ходе всего проекта, а также на фазе приемки - для подтверждения результата.

Приоритет проекта определяется на основе оценки трех показателей:

 Финансовая ценность.

 Стратегическая ценность.

 Уровень рисков.

**3. Процессы управления проектом**

**Под управлением проектом** понимается деятельность, направленная на последовательное достижение ожидаемых результатов в условиях трех ограничений. Она требует интегрального управления многими процессами, их взаимодействием, поиска компромиссов.

Сегодня управление проектами вылилось в самостоятельную дисциплину со своими стандартами, методикой, сводом знаний. Она рассматривает процессы, общие для всех проектов и независящие от предметных областей. На рисунке 5.4 показаны процессы и основные функции управления проектами.



Рисунок 5.4. Процессы и основные функции управления проектами

С целью структурирования **процессы управления проектом** принято делить на области знаний.

1. Управление интеграцией.

2. Управление содержанием.

3. Управление сроками.

4. Управление стоимостью.

5. Управление качеством.

6. Управление рисками.

7. Управление человеческими ресурсами.

8. Управление коммуникациями.

9. Управление конфигурацией.

***1. Управление интеграцией*** включает в себя процессы и действия, необходимые для определения, уточнения, комбинирования, объединения и координирования различных процессов и действий по управлению проектом в рамках групп процессов управления проектом. Таким образом, целью данного процесса является достижение эффективного взаимодействия процессов управления проектами, обеспечивающих достижение целей проекта. Процессы, входящие в состав управления интеграцией:

● Разработка Технико-экономического обоснования (ТЭО) проекта.

● Разработка устава проекта.

● Разработка плана управления проектом.

● Руководство и управление исполнением проекта.

● Осуществление интегрированного управления изменениями.

● Оценка альтернатив развития проекта.

● Планирование закрытия проекта и перехода в стадию эксплуатации.

● Завершение проекта.

***2. Управление содержанием*** включает в себя процессы и действия, обеспечивающие включение в проект всех тех и только тех работ, которые необходимы для успешного выполнения проекта. Оно непосредственно связано с определением и контролем того, что включено или не включено в проект. Процессы, входящие в состав управления содержанием:

● Формирование требований проекта.

● Формирование ИСР.

● Определение содержания проекта.

● Оценка реализуемости требований проекта.

● Подтверждение содержания проекта.

● Определение уточненных системных требований.

● Мониторинг содержания и объема проекта.

● Оценка готовности пользователей к работе в системе.

● Планирование обучения конечных пользователей.

***3. Управление сроками*** проекта включает в себя процессы, обеспечивающие своевременное завершение проекта. Процессы, входящие в состав управления сроками проекта:

● Формирование списка работ проекта.

● Определение последовательности работ проекта.

● Оценка трудоемкости и продолжительности работ.

● Разработка базового расписания проекта.

● Контроль и управление расписанием проекта.

***4. Управление стоимостью*** проекта объединяет процессы, выполняемые в ходе планирования, разработки бюджета и контролирования затрат и обеспечивающие завершение проекта в рамках утвержденного бюджета. Процессы, входящие в состав управления стоимостью проекта:

● Оценка стоимости проекта.

● Разработка сметы проекта.

● Разработка базового плана по стоимости.

● Управление стоимостью проекта.

5. Процессы ***управления качеством*** проекта объединяют все осуществляющиеся в исполняющей организации операции, определяющие политику, цели и распределение ответственности в области качества таким образом, чтобы проект удовлетворял тем нуждам, для которых он был предпринят.

Управление качеством осуществляется посредством системы управления, предусматривающей определенные правила, процедуры и процессы по планированию качества, обеспечению качества и контролю качества, а также операции по их совершенствованию. К процессам управления качеством проекта относятся:

● Формирование программы качества проекта.

● Формирование базовой линии требований проекта.

● Управление требованиями проекта.

● Осуществление обеспечения качества.

● Тестирование.

● Приемка результатов.

6. Процесс ***управления рисками*** тесно связан с общим жизненным циклом проекта. На ранних этапах преобладают риски, связанные с бизнесом, рамками проекта, требованиями к конечному продукту и проектированием этого продукта. На стадии реализации доминируют технологические риски, далее возрастает роль рисков, связанных с поддержкой и сопровождением системы. На протяжении всего жизненного цикла проекта возникают новые риски, что требует проведения дополнительных операций анализа и планирования.

Согласно ГОСТ Р ИСО/МЭК 152888-2005 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 проводятся 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. К процессам управления рисками проекта относятся:

● Планирование управления рисками.

● Идентификация рисков.

● Качественный анализ рисков.

● Количественный анализ рисков.

● Планирование реагирования на риски.

● Мониторинг и управление рисками.

***7. Управление человеческими ресурсами*** проекта – это процесс обеспечения эффективного использования человеческих ресурсов проекта, к которым относятся все участники проекта (спонсоры, заказчики, команда проекта, субподрядчики, подразделения компании и другие участники проекта). К процессам управления человеческими ресурсами проекта относятся:

● Планирование человеческих ресурсов.

● Набор команды проекта.

● Оценка доступности.

● Развитие и оценка команды проекта.

● Организация инфраструктуры проекта.

***8. Управление коммуникациями*** проекта – это процессы идентификации и эффективного обеспечения всех участников проекта информацией о проекте, а также создания единого образа проекта внутри организации. К ним относятся:

● Идентификация участников проекта.

● Формирование стратегии и плана коммуникаций.

● Реализация плана коммуникаций и сбор обратной связи.

***9. Управление конфигурацией*** – процесс управления аппаратными средствами, программным обеспечением, данными, а также документацией в ходе разработки, тестирования и использования информационных систем. Цель процесса управления конфигурацией состоит в установлении и поддержании целостности всех идентифицированных выходных результатов проекта или процесса обеспечения доступа к ним любой заинтересованной стороны. К процессам управления конфигурацией проекта относятся:

● Идентификация объектов управления конфигурацией.

● Планирование инфраструктуры стадии разработки.

● Установление базовой линии конфигурации проекта.

● Оценка соответствия базовой линии конфигурации.

● Контроль конфигурации выделенных элементов проекта.

● Обеспечение целостности элементов конфигурации.

● Реконфигурация инфраструктуры проекта.

***Основная литература***

1. Архипенков С. Лекции по управлению программными проектами// Электронный ресурс:http://citforum.ru/SE/project/

2. Основы управления ИТ-проектами [Электронный ресурс]: учебное пособие / сост. Е.Р. Кирколуп, Ю.Г., Скурыдин, Е.М. Скурыдина. – Барнаул : АлтГПУ, 2017// Электронный ресурс:http://library.altspu.ru/dc/pdf/kirkolup.pdf

3. Рыбалова Е.А. Управление проектами: учебное пособие / Е.А. Рыбалова. - Томск: ФДО, ТУСУР, 2015. - 205 с.

4. Богданов В.В. Управление проектами в Microsoft Project 2007. Учебный курс. -  СПб: Питер, 2010

[5. Коптелов А., Никитин И., Цулая М. Гибкая процессная методология Agile](https://www.intuit.ru/studies/courses/3590/832/info). - М.: ИНТУИТ // Электронный ресурс: // www.intuit.ru/studies/courses/3590/832/print\_lecture/31170

6. Топ-7 методов управления проектами: Agile, Scrum, Kanban, PRINCE2 и другие// www.pmservices.ru/project-management-news/top-7-metodov-upravleniya-proektami-agile-scrum-kanban-prince2-i-drugie/

### 7. [Грекул](https://www.intuit.ru/intuituser/userpage/v_i_grekul) В., [Коровкина](https://www.intuit.ru/intuituser/userpage/n_l_korovkina) Н., [Куприянов](https://www.intuit.ru/intuituser/userpage/y_v_kupriyanov) Ю. [Методические основы управления ИТ-проектами](https://www.intuit.ru/studies/courses/646/502/info). - М.: ИНТУИТ // Электронный ресурс: https://www.intuit.ru/studies/courses/646/502/info

### ***Дополнительная литература:***

8. Сериков А. Основы управления проектами. - М.: ИНТУИТ. Электронный ресурс: www.intuit.ru/studies/professional\_retraining/965/courses/272/info

9. Управление проектами: Учебное пособие / М.В. Романова. - М.: ИД ФОРУМ: НИЦ ИНФРА-М, 2014

10. Руководство к Своду знаний по управлению проектами РМВОК (A Guide to the Project Management Body of Knowledge PMBOK®) — 4-е изд. — Project Management Institute, Inc., 2008. — 464 с.

11. https://products.office.com/

**Лекция 6. Планирование проекта**

*План:*

1. План управления проектом

2. Формирование иерархической структуры проекта

3. Планирование управления содержанием

4.Планирование организационной структуры

### 5. Планирование управления конфигурациями

6. Планирование управления качеством

**1. План управления проектом**

Процесс разработки плана управления проектом есть процесс документации действий, необходимых для определения, подготовки, интеграции и координации всех вспомогательных планов. Корректно составленный план управления проектом является основным источником информации о том, как проект будет планироваться, оцениваться, контролироваться и закрываться.

План управления проектом обновляется и редактируется в рамках процесса осуществления интегрированного управления изменениями проекта, для поддержки версионности документа рекомендуется использовать лист управления документом. План управления проектом может быть либо резюмирующим, либо детализированным и состоять из одного или нескольких вспомогательных планов и прочих элементов.

План управления проектом рекомендуется разделять на 3 блока по характеру содержащейся в них информации:

1. Вспомогательные планы управления проектом, в число которых входят:

● план управления содержанием проекта;

● план управления расписанием проекта;

● план управления стоимостью проекта;

● план управления качеством проекта;

● план управления обеспечением персоналом;

● план управления коммуникациями проекта;

● план управления рисками проекта;

● план управления конфигурацией.

2. Базовая линия проекта, состоящая из:

● базового расписания проекта;

● базового плана по стоимости;

● базового плана по качеству;

● базового плана по конфигурации;

● реестра рисков.

3. Результаты анализа, проведенного проектной командой в отношении содержания, объема и сроков проекта.

Процессы, используемые для управления содержанием проекта, а также вспомогательные инструменты и методы различаются в зависимости от прикладной области и обычно определяются как часть жизненного цикла проекта. Одобренное подробное описание содержания проекта вместе с ИСР и словарем ИСР представляют собой *базовый план проекта по содержанию*. Далее *содержание, оформленное в базовом плане*, отслеживается, подтверждается и контролируется на всем протяжении жизненного цикла проекта.

**2. Формирование иерархической структуры проекта**

Человечество пока не придумало ничего более эффективного для решения сложной задачи, чем анализ и ее декомпозиция на более простые подзадачи, которые, в свою очередь, могут быть разделены на еще более простые подзадачи и так далее. Получается некоторая иерархическая структура, дерево, в корне которого находится проект, а на листьях элементарные задачи или работы, которые надо выполнить, чтобы завершить проект в условиях заданных ограничений.

**Иерархическая структура работ (ИСР)** (**Work Breakdown Structure, WBS**) - ориентированная на результат иерархическая декомпозиция работ, выполняемых командой проекта для достижения целей проекта и необходимых результатов. С ее помощью структурируется и определяется все содержание проекта. Каждый следующий уровень иерархии отражает более детальное определение элементов проекта.

**Создание иерархической структуры работ**— это процесс разделения (***декомпозиция***) результатов проекта и работ по проекту на более мелкие элементы, которыми легче управлять.

Основой для разработки ИСР служит концепция проекта, которая определяет продукты проекта и их основные характеристики. ***ИСР обеспечивает выявление всех работ, необходимых для достижения целей проекта.***

Многие проекты проваливаются не от того, что у них нет плана, а от того что в этом плане забыты важные работы, например, тестирование и исправление ошибок, и продукты проекта, например пользовательская документация. Поэтому, если ИСР составлена корректно, то любая работа, которая в нее не вошла не может считаться работой по проекту.

ИСР делит проект на *подпроекты, пакеты работ, подпакеты*. Каждый следующий уровень декомпозиции обеспечивает последовательную детализацию содержания проекта, что позволяет производить оценку сроков и объемов работ. ИСР должна включать все промежуточные и конечные продукты.

Выполнять декомпозицию работ проекта можно по-разному.   
Например, ГОСТ 19.102-77 предусматривает каскадный подход и определяет следующие стадии разработки программной системы:

1. Техническое задание
2. Эскизный проект
3. Технический проект
4. Рабочий проект
5. Внедрение

Если следовать этому стандарту, то на первом уровне ИСР должны находиться именно эти проектные продукты. Если бы пришлось разрабатывать АСУ для управления ядерным реактором или пилотируемым космическим аппаратом, то именно так и следовало поступать.   
Однако в коммерческой разработке ПО такой подход не эффективен. Как мы уже говорили, *современный процесс разработки коммерческого ПО должен быть****инкрементальным****.* Это означает, что на верхнем уровне декомпозиции нашего проекта должны находиться продукты проекта, а на следующем уровне — компоненты, из которых эти продукты состоят. Компоненты далее могут быть декомпозированы на «фичи» — функции, которые они должны реализовывать.

Выделение компонентов, составляющих программный продукт, это элемент высокоуровневого проектирования, которое мы должны выполнить на фазе планирования проекта, не дожидаясь проработки всех функциональных требований к разрабатываемому ПО.

Компонентами могут быть как прикладные подсистемы, так и инфраструктурные или ядерные, например, подсистема логирования, безопасности, библиотека визуальных компонентов GUI.

При составлении базового плана работ не стоит стремиться максимально детализировать все работы. ИСР не должна содержать слишком много уровней, достаточно 3-5.

***Пример.*** Например, ИСР нашего проекта-примера разработки «Автоматизированной системы продажи документации» может выглядеть следующим образом. Иерархическая структура работ проекта разработки «Автоматизированной системы продажи документации» (*курсивом выделены контрольные точки проекта*)

1. Проект разработки «Автоматизированной системы продажи документации»   
1.1. Подготовка технического задания на автоматизацию

1.1.1.1. Проведение аналитического обследования   
1.1.1.2. Разработка функциональных требований   
1.1.1.3. Разработка требований базовому ПО   
1.1.1.4. Разработка требований к оборудованию и к операционно-системному ПО   
1.1.1.5. Согласование и утверждение ТЗ   
*1.1.1.6. ТЗ утверждено*

1.2. Поставка и монтаж оборудования

1.2.1.Разработка спецификации на оборудование   
1.2.2.Закупка и поставка оборудования   
1.2.3. Монтаж оборудования   
1.2.4. Установка и настройка операционно-системного ПО   
*1.2.5. Монтаж оборудования завершен*

1.3. Поставка и установка базового ПО

1.3.1 .Разработка спецификаций на базовое ПО   
1.3.2.Закупка базового ПО   
1.3.3. Развертывание и настройка базового ПО   
*1.3.4. Базовое ПО установлено у заказчика*

1.4. Разработка и тестирование прикладного ПО

1.4.1. Разработка спецификаций на прикладное ПО   
1.4.2. Установка и конфигурирование рабочей среды   
1.4.3. Проектирование и разработка ПО

1.4.3.1. Авторизация и аутентификация пользователей.   
1.4.3.2. Разработка подсистемы заказа документации   
1.4.3.2.1. Просмотр каталога продуктов.   
1.4.3.2.2. Поиск продуктов по каталогу.   
1.4.3.2.3. Заказ выбранных продуктов.   
1.4.3.2.4. Просмотр информации о статусе заказа.   
1.4.3.2.5. Информирование клиента об изменении статуса заказа.   
*1.4.3.2.6. Подсистема заказа документации передана в тестовую эксплуатацию (на серверах разработчика).*   
1.4.3.3. Разработка подсистемы обработки заказов   
1.4.3.3.1. Просмотр и обработка заказов исполнителями из службы продаж.   
1.4.3.3.2. Просмотр статистики поступления и обработки заказов за период.   
*1.4.3.3.3. Подсистема обработки заказов передана в тестовую эксплуатацию на оборудовании Заказчика*   
1.4.3.4. Разработка подсистемы сопровождения каталога   
1.4.3.4.1. Подготовка и сопровождение каталога продукции.   
1.4.3.5. Исправление ошибок

1.4.4. Тестирование ПО

1.4.4.1. Раунд 1   
1.4.4.2. Раунд 2   
1.4.4.3. Раунд 3   
1.4.4.4. Выходное тестирование

1.4.5. Документирование прикладного ПО

1.5. Обучение пользователей

1.5.1 .Подготовка учебных курсов

1.5.2.Обучение сотрудников Отдела 123   
1.5.3.Обучение руководства ОАО XYZ  
1.5.4.Обучение администраторов системы

1.6. Ввод в опытную эксплуатацию

1.6.1. Развертывание и настройка прикладного ПО   
1.6.2. Проведение приемо-сдаточных испытаний   
*1.6.3. Акт передачи системы в опытную эксплуатацию утвержден*

1.7. Сопровождение системы в период опытной эксплуатации   
*1.8. Система передана в промышленную эксплуатацию*

Должна быть установлена персональная ответственность за все части проекта (подпроекты и пакеты работ). Для каждого пакета работ должен быть четко определен результат на выходе. Работы и оценки проекта должны быть согласованы с ключевыми участниками команды, руководством компании-исполнителя и, при необходимости, с заказчиком.

В результате согласования члены команды принимают на себя обязательства по реализации проекта, а руководство принимает на себя обязательства по обеспечению проекта необходимыми ресурсами.

ИСР является одним из основных инструментов (средств) в механизме управления проектом, с помощью которого измеряется степень достижения результатов проекта. Важнейшая ее функция - это обеспечить консистентное представление всех у частников проекта относительно того, как будет делаться проект.

В последующем базовый план будет служить ориентиром для сравнения с текущим исполнением проекта и выявления отклонений для целей управления.

Существуют два основных способа разработки ИСР: «сверху вниз» и «снизу вверх». Приведем **описание подхода «сверху вниз».**

1. Сбор исходной информации.

Разработка ИСР станет более легким и осмысленным делом, если будет доступна следующая информация:

● требования заказчика;

● пул доступных ресурсов;

● конкретная проектная ситуация.

2. Выбор типа ИСР.

После получения необходимой информации о факторах, влияющих на ИСР, необходимо определиться с ее типом построения:

● по жизненному циклу,

● по системам,

● по географическим зонам.

3. Определение степени детализации ИСР.

Принимая во внимание тот факт, что число пакетов влияет на время и стоимость управления проектом, нужно выбрать такое количество пакетов работ, для управления которыми есть время и бюджет. Пакетом работ называется основной элемент управления ИСР, дискретная задача, имеющая определимые конечные результаты, за достижение которых отвечают организационные единицы. Очевидно, пакеты работ должны представлять небольшие результаты и быть управляемыми.

Для определения степени детализации ИСР нужна следующая информация:

● количество уровней в ИСР;

● количество и средний размер пакета работ, принятые в отрасли.

### **3. Планирование управления содержанием**

Одна из распространенных «болезней» программных проектов называется «ползучий фичеризм». Это, когда к изначально спроектированной будке для любимой собаки сначала пристраивают сарайчик для хранения садового инвентаря, а потом и домик в несколько этажей для ее хозяина. И все это пытаются построить на одном и том же фундаменте и из тех же самых материалов. Эта болезнь стала причиной летального исхода многих проектов разработки ПО.

Поэтому, сразу, как только удалось стабилизировать и согласовать ИСР, необходимо разработать план управления содержанием проекта. Для этого следует:

* Определить источники запросов на изменение.
* Установить порядок анализа, оценки и утверждения/отклонения изменения содержания.
* Определить порядок документирования изменений содержания.
* Определить порядок информирования об изменении содержания.

Первая задача, которую необходимо решить при анализе запроса на изменения — выявить объекты изменений: требования, архитектура, структуры данных, исходные коды, сценарии тестирования, пользовательская документация, проч. Затем требуется спроектировать и детально описать изменения во всех выявленных объектах. И наконец, следует оценить затраты на внесение изменений, тестирование изменений и регрессионное тестирование продукта и их влияние на сроки проекта.

Эта работа, которая потребует затрат рабочего времени и порой значительных разных специалистов: аналитиков, проектировщиков, разработчиков, тестировщиков, наконец, менеджера проекта. Поэтому эта работа должна обязательно быть учтена в плане.

### 4.**Планирование организационной структуры**

**Организационная структура** - это согласованное и утвержденное распределение ролей, обязанностей и целей деятельности ключевых участников проекта. *Она в обязательном порядке должна включать в себя систему рабочих взаимоотношений между рабочими группами проекта, систему отчетности, оценки хода выполнения проекта и систему принятия решений*. Следует помнить, что организационная структура проекта — «живой» организм. Она начинает складываться на стадии планирования и должна меняться по ходу проекта.  
И еще. *Нестабильность организационной структуры — частая смена исполнителей — может стать серьезной проблемой в управлении проектом,* поскольку, существует цена замены, которая определяется временем вхождения нового участника в контекст проекта.

### **5. Планирование управления конфигурациями**

Конфигурационное управление один из важных процессов производства программного обеспечения.

План проекта должен включать в себя работы по обеспечению единого хранилища всей проектной документации и разрабатываемого программного кода, обеспечению сохранности и восстановление проектной информации после сбоя. Работы по настройке рабочих станций и серверов, используемых участниками проектной команды, тоже должны войти в план. Кроме этого в плане должны содержаться работы, необходимые для организации сборки промежуточных выпусков системы, а также ее конечного варианта.  
Эти работы, как правило, выполняет один человек — инженер по конфигурациям.

Управление конфигурациями может многократно усложниться, если проектной команде параллельно с разработкой новой функциональности продукта приходится поддерживать несколько релизов этого продукта, которые были установлены ранее у разных клиентов. Все эти работы должны быть учтены в плане проекта.

### **6. Планирование управления качеством**

Обеспечение качества еще одна из базовых областей знаний в программной инженерии. Обеспечение качества это важная работа, которая должна быть спланирована заранее и выполняться по ходу всего программного проекта, а не только во время приемо-сдаточных испытаний.

При планировании этой работы необходимо понимать, что продукт проекта не должен обладать наивысшим возможным качеством, которое недостижимо за конечное время. Необходимое качество продукта определяется требованиями к нему.   
Основная задача обеспечения качества это не поиск ошибок в готовом продукте (выходной контроль) а их предупреждение в процессе производства.   
**План управления качеством** должен включать в себя следующие работы:

* Объективную проверку соответствия программных продуктов и технологических операций применяемым стандартам, процедурам и требованиям.
* Определение отклонений по качеству, выявление их причин, применение мер по их устранению, а также контроль исполнения принятых мер и их эффективности.
* Представление высшему руководству независимой информации о несоответствиях, не устраняемых на уровне проекта.

Помимо перечисленных разделов план проекта должен включать еще:

* План управления рисками
* Оценку трудоемкости и сроков работ

После определения трудоемкости работ необходимо определить график их выполнения и общие сроки реализации проекта — составить расписание работ по проекту.

### **Выводы**

На верхнем уровне ИСР должны находиться не процессы, а продукты проекта, на следующем уровне — компоненты из которых эти продукты состоят. Выделение компонентов, составляющих программный продукт, это элемент высокоуровневого проектирования, которое мы должны выполнить на фазе планирования проекта, не дожидаясь проработки всех функциональных требований к разрабатываемому ПО.

Помимо работ, непосредственно направленных на создание программного обеспечения, в плане проекта должны быть предусмотрены необходимые ресурсы для обеспечения работ по следующим процессам:

* управление содержанием;
* управление конфигурациями,
* управление качеством,
* управление рисками,
* управление проектом.

В проекте всегда существует хотя бы один критический путь, но их может быть несколько. Критический путь может меняться во время исполнения проекта. При исполнении проекта руководитель должен обращать внимание на исполнение задач на критическом пути в первую очередь и следить за появлением других критических путей.

***Основная литература***

1. Архипенков С. Лекции по управлению программными проектами// Электронный ресурс:http://citforum.ru/SE/project/

2. Основы управления ИТ-проектами [Электронный ресурс]: учебное пособие / сост. Е.Р. Кирколуп, Ю.Г., Скурыдин, Е.М. Скурыдина. – Барнаул : АлтГПУ, 2017// Электронный ресурс:http://library.altspu.ru/dc/pdf/kirkolup.pdf

3. Рыбалова Е.А. Управление проектами: учебное пособие / Е.А. Рыбалова. - Томск: ФДО, ТУСУР, 2015. - 205 с.

4. Бухарев Н.Р. Введение в гибкое управление проектами. Краткий конспект лекций// tulpar.kfu.ru/course/view.php?id=2284

[5. Коптелов А., Никитин И., Цулая М. Гибкая процессная методология Agile](https://www.intuit.ru/studies/courses/3590/832/info). - М.: ИНТУИТ // Электронный ресурс: // www.intuit.ru/studies/courses/3590/832/print\_lecture/31170

6. Топ-7 методов управления проектами: Agile, Scrum, Kanban, PRINCE2 и другие// www.pmservices.ru/project-management-news/top-7-metodov-upravleniya-proektami-agile-scrum-kanban-prince2-i-drugie/

### 7.[Грекул](https://www.intuit.ru/intuituser/userpage/v_i_grekul) В., [Коровкина](https://www.intuit.ru/intuituser/userpage/n_l_korovkina) Н., [Куприянов](https://www.intuit.ru/intuituser/userpage/y_v_kupriyanov) Ю. [Методические основы управления ИТ-проектами](https://www.intuit.ru/studies/courses/646/502/info). - М.: ИНТУИТ // Электронный ресурс: https://www.intuit.ru/studies/courses/646/502/info

### ***Дополнительная литература:***

8. Сериков А. Основы управления проектами. - М.: ИНТУИТ. Электронный ресурс: www.intuit.ru/studies/professional\_retraining/965/courses/272/info

9. Управление проектами: Учебное пособие / М.В. Романова. - М.: ИД ФОРУМ: НИЦ ИНФРА-М, 2014

10. Руководство к Своду знаний по управлению проектами РМВОК (A Guide to the Project Management Body of Knowledge PMBOK®) — 4-е изд. — Project Management Institute, Inc., 2008. — 464 с.

11. https://products.office.com/

**Лекция 7. Разработка расписания проекта**

*План:*

1. Управление сроками проекта

2. Расписание проекта

### 3. Базовое расписание проекта

**1. Управление сроками проекта**

*Управление сроками* проекта включает в себя процессы, обеспечивающие своевременное завершение проекта. Общая схема процессов управления сроками проекта включает следующие:

1. *Определение операций* — процесс определения конкретных операций, которые необходимо выполнить для получения результатов проекта.

2. *Определение последовательности операций* — процесс выявления и документирования зависимостей между операциями проекта.

3. *Оценка ресурсов операций* — процесс оценки типов и количества материалов, человеческих ресурсов, оборудования или поставок, необходимых для выполнения каждой операции.

4. *Оценка длительности операций* — процесс приблизительного определения количества рабочих периодов, требуемых для завершения отдельных операций при предполагаемых ресурсах.

5. *Разработка расписания* — процесс анализа последовательностей операций, их длительности, потребности в ресурсах и временных ограничений для создания расписания проекта.

6. *Управление расписанием* — процесс мониторинга статуса проекта для корректировки его исполнения и внесения изменений в базовое расписание.

Данные процессы взаимосвязаны друг с другом, а также с процессами из других областей знаний. К наиболее известным методологиям составления расписания относятся *методы критического пути* и критической цепи.

**2. Расписание проекта**

**Разработка расписания проекта – итеративный процесс, определяющий плановые даты начала и завершения операций проекта**. Разработка расписания производится непрерывно по мере выполнения работ проекта. При этом могут потребоваться проверка и редактирование оценки длительности и ресурсов, чтобы в итоге получить одобренное расписание проекта. **Согласованное расписание используется как базовое, по которому будет оцениваться прогресс рисков.**

**Исходной информацией для процесса разработки расписания является описание содержания проекта**. Оно включает допущения (документированные факторы, относящиеся к расписанию, которые при разработке расписания считаются достоверными) и ограничения (факторы, ограничивающие свободу выбора команды управления проектом при проведении анализа сети расписания и влияющие на составление расписания проекта).

При разработке расписания учитываются два основных типа ограничений по времени:

● требуемые даты для начала или завершения операции, которые можно использовать для ограничения начала или завершения операции;

● контрольные события, вследствие чего получение определенных результатов работ привязывается к определенным датам, изменить которые можно только посредством одобренных изменений.

Для разработки расписания рекомендуется использовать следующие инструменты и методы.

***Инструменты и методы составления расписания***

*Анализ сети* представляет собой *технологию* создания расписания проекта.

Применяются разнообразные аналитические методы, такие как:

- *метод критического пути*,

- *метод критической цепи,*

*- анализ сценариев* «*что если*»

- *выравнивание* ресурсов, позволяющие рассчитать даты раннего и позднего старта и финиша незавершенных операций проекта.

Некоторые пути в сети могут иметь точки слияния или расхождения, их можно выявить и использовать в анализе *сжатия расписания* и других видах анализа.

1. ***Метод критического пути***позволяет рассчитать теоретические даты раннего и позднего старта и финиша для всех операций без учета ресурсных ограничений, путем проведения анализа *прохода вперед и назад* по сети проекта. Полученные даты не обязательно являются расписанием проекта. Они скорее указывают периоды времени, в рамках которых могут быть запланированы операции с учетом их длительностей, логических связей, опережений, задержек и других известных *ограничений*.

Для любого пути в сети ***гибкость* расписания**, называемая «*полным временным резервом*», измеряется положительной разницей между ранними и поздними датами. ***Критический путь***характеризуется нулевым полным временным резервом. ***Свободный* временной резерв** — период времени, на который операция может быть отложена, не вызывая задержки раннего старта любой непосредственно последующей операции в данном сетевом пути.

2. ***Метод критической цепи***— метод анализа сети, который изменяет расписание проекта с учетом *ограниченности* ресурсов. Изначально сетевая диаграмма проекта строится на основе оценок длительности, заданных зависимостей и ограничений. Затем рассчитывается критический путь, учитывается наличие ресурсов и в результате определяется расписание с учетом ресурсных ограничений. Полученное расписание часто имеет измененный критический путь.

Критический путь с ресурсными ограничениями известен как «*критическая цепь*». **Метод *критической цепи* добавляет буферы длительности в виде операций, не предусматривающих выполнения работ, для управления неопределенностью.**

Один из буферов, расположенный в конце критической цепи, известен как *проектный буфер* и защищает статусную дату завершения от задержек на критической цепи.

3. *Анализ сценариев «что если»* — анализ вопроса: «Что произойдет, если ситуация будет развиваться по сценарию «*X*»?» Выполняется анализ сети, при котором с помощью модели расписания просчитываются различные сценарии (например, задержка поставки основных элементов, увеличение длительности отдельных операций) или моделируется влияние непредвиденных внешних факторов (например, забастовка или изменение процедуры лицензирования). Результаты анализа могут использоваться для оценки выполнимости расписания проекта в условиях *неопределенности и риска*, для составления резервных планов реагирования, преодоления или смягчения неблагоприятных последствий. Моделирование включает в себя расчет различных длительностей проекта. Наиболее известен *метод Монте-Карло*, в котором распределение вероятных значений длительности определяется для каждой операции и используется для вычисления распределения вероятных выходов всего проекта.

4. *Выравнивание ресурсов* представляет собой *метод анализа сети*, применяемый для расписания, которое уже было проанализировано методом критического пути. Выравнивание ресурсов необходимо при отклонениях от заданной доступности необходимых ресурсов на параллельных операциях.

5. *Применение опережений и задержек* — это уточнения, вносимые во время анализа сети для разработки жизнеспособного расписания.

6. *Сжатие расписания* сокращает длительность проекта без изменения содержания проекта, временных ограничений, статусных дат или иных целевых параметров расписания. Методы сжатия расписания включают:

• *сжатие* — анализируются компромиссы между стоимостью и расписанием, чтобы определить, как возможно максимально сжать сроки при минимальных затратах. Примеры сжатия могут включать одобрение сверхурочной работы, использование дополнительных ресурсов или плату за ускорение поставки для операций на критическом пути;

• *быстрый проход* — сжатие расписания происходит за счет параллельного выполнения фазы или операции, обычно выполняемых последовательно.

Примером является строительство фундамента здания до подготовки всех архитектурных чертежей. Быстрый проход применим только в том случае, когда операции могут накладываться одна на другую для сокращения длительности.

*Инструмент составления расписания* — автоматические инструменты составления расписания облегчают процесс составления расписания, генерируя даты старта и финиша на основе информации об операциях, сетевых диаграммах, ресурсах и длительностях операций.

**Диаграмма, построенная по методу критического пути** – методу анализа сети расписания, проводимого при помощи модели расписания. Критический путь представляет группу операций, которые не могут быть задержаны без изменения отсрочки, даты завершения всего проекта. При использовании метода критического пути рассчитываются теоретические даты раннего старта и раннего финиша и позднего старта и позднего финиша для всех плановых операций без учета ограничений по ресурсам. Этот расчет производится путем проведения анализа прямого и обратного проходов по путям сети расписания проекта. Полученные даты раннего и позднего старта и финиша показывают периоды времени, в пределах которых следует планировать данную операцию, исходя из ее длительности, логических взаимосвязей, опережений, задержек и прочих ограничений.

***Расписание проекта***содержит, по меньшей мере, плановую дату старта и плановую дату финиша для каждой операции. Расписание проекта может быть разработано детально или укрупнено как расписание контрольных событий.

***Результатами процесса разработки расписания являются:***

● *Данные для модели расписания*

● *Базовый план расписания*

Расписание проекта может быть представлено в обобщенном виде, «*расписанием контрольных событий*», или в подробном виде в форме таблицы, но чаще используется графическое представление в одном из *следующих форматов*:

• ***Диаграммы контрольных событий***аналогичны ленточным диаграммам, но *показывают только запланированные даты начала или завершения операции и ключевые внешние события*.

Диаграмма контрольных событий– инструмент для разработки расписания проекта, построение которого включает следующие действия:

● сбор исходной информации для построения диаграммы;

● построение сетевой диаграммы, отражающей взаимосвязь операций;

● определение уровня детализации контрольных событий – количества контрольных событий, отражаемых на диаграмме;

● выбор контрольных событий, которые являются главными для продвижения проекта;

● упорядочивание контрольных событий – изучение взаимосвязей и определение последовательности их выполнения;

● нанесение контрольных событий на детальное расписание проекта;

● проверка равномерности распределения контрольных событий по расписанию проекта.

• *Ленточные диаграммы* (рис. 7.1), в которых полоски представляют *операции*, показывают даты начала и завершения операций и их ожидаемые длительности. Диаграмма Гантта– диаграмма, которая использует горизонтальные полосы для представления операций проекта, показывает даты начала и завершения каждой операции и проекта относительно горизонтальной шкалы времени.

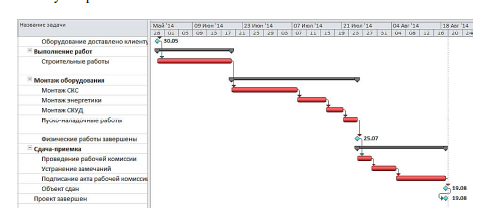


Рисунок 7.1. Ленточная диаграмма Gantt

• *Сетевые диаграммы проекта* содержат информацию о датах операций, обычно показывают как логику сети проекта, так и операции критического пути проекта.

● *Данные для модели расписания.*Обязательные данные для расписания проекта включают в себя контрольные события расписания, плановые операции, параметры операции и документацию всех имеющихся допущений и ограничений, а дополнительные – требования к ресурсам по периодам времени, альтернативные расписания, резервы на непредвиденные обстоятельства.

● *Базовый план расписания* **–** особый вариант расписания проекта, разрабатываемый посредством анализа сети расписания модели расписания, принимается и утверждается командой управления проектом в качестве первоначального (базового) плана расписания с указанными базовым стартом и базовым финишем (рисунок 7.2).

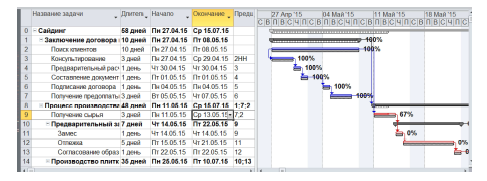


Рисунок 7.2. Базовое расписание выполняемого проекта на текущую дату

Базовый план расписания используют для выявления отклонений фактических сроков выполнения операций от плановых.

● *Требования к ресурсам.*

● *Параметры операции.*

● *Календарь проекта.*

● *Запрошенные изменения.*В процессе разработки расписания могут появиться запрошенные изменения, которые обрабатываются в процессе общего управления изменениями.

● *План управления проектом.*План управления проектом обновляется.

*Управление расписанием* включает:

• определение текущего состояния расписания проекта;

• влияние на факторы, вызывающие изменения расписания;

• определение фактов изменения расписания проекта;

• управление фактическими изменениями по мере их возникновения.

Управление расписанием является элементом процесса осуществления общего управления изменениями.

### 3. **Базовое расписание проекта**

После определения трудоемкости работ необходимо определить график их выполнения и общие сроки реализации проекта — составить расписание работ по проекту.

***Базовое расписание***— утвержденный план-график с указанными временными фазами проекта, контрольными точками и элементами иерархической структуры работ.

*Базовое расписание может быть наиболее наглядно представлено диаграммой Ганта.* В этой диаграмме плановые операции или элементы иерархической структуры работ перечислены с левой стороны, даты отображаются сверху, а длительность операций показана горизонтальными полосками от даты начала до даты завершения.  
Базовое расписание это, как правило, элемент контракта с заказчиком. **Контрольные точки (вехи)** должны служить точками анализа состояния проекта и принятия решения «GO/NOT GO», поэтому они должны зримо демонстрировать статус проекта. Контрольная точка «Проектирование завершено» — плохо. Наиболее эффективный подход — метод последовательных поставок: контрольная точка «Завершено тестирование требований 1, 3, 5, 7»

*Если работы не связаны между собой, то любую из них мы можем начинать и завершать, когда нам удобно. Все работы можно делать параллельно и, в этом случае, минимальная длительность проекта равна длительности самой долгой работы.* Однако на практике между работами существуют зависимости, которые могут быть «жесткими», например, анализ — проектирование — кодирование — тестирование и документирование конкретной функции; или «нежесткими», которые могут пересматриваться или смягчаться.

Например, последовательное выполнение задач конкретным исполнителем (можно перепланировать на другого исполнителя) или разработка базового ПО, которая должна предшествовать разработке прикладного ПО. В этом случае можно создавать «заглушки» эмулирующие работу базового ПО. Таким образом, диаграмма Ганта для расписания проекта выглядит как гамак, составленный из множества цепочек взаимосвязанных работ с единой точкой начала и завершения.

***Критический путь проекта*** (Critical path) — самая длинная цепочка работ в проекте. Увеличение длительности любой работы в этой цепочки приводит к увеличению длительности всего проекта.

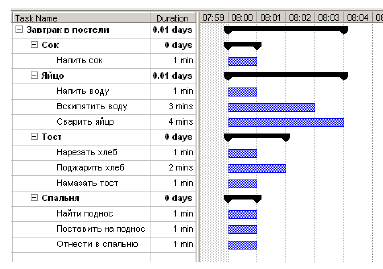
В проекте всегда существует хотя бы один критический путь, но их может быть несколько. Критический путь может меняться во время исполнения проекта. При исполнении проекта руководитель должен обращать внимание на исполнение задач на критическом пути в первую очередь и следить за появлением других критических путей.

Практическая рекомендация: на критическом пути должны стоять работы с нежесткими связями, которые всегда можно перепланировать, если возникает угроза срыва сроков.

**Пример.** Чтобы проиллюстрировать понятие критического пути рассмотрим пример «суперпроекта»

**Концепция проекта** выглядит следующим образом.  
Цель проекта. Сделать завтрак в постель  
Результаты проекта. Завтрак в постели из вареного яйца, тоста и апельсинового сока.  
Ресурсы. Имеется один оператор и обычное кухонное оборудование.  
Сроки. Проект начинается на кухне в 8:00 и завершается в спальне.  
Критерий приемки. Используются минимальные трудовые ресурсы и срок. Конечный продукт имеет высокое качество: яйцо свежесваренное, тост теплый, сок холодный.  
Обоснование полезности. Проект служит достижению стратегических целей.

Иерархическая структура работ, ориентированная на конечный продукт, с оценкой их длительности представлена на Рисунке 7.3.

   
Рисунок 7.3. Иерархическая структура работ «суперпроекта»

**На следующем шаги мы должны учесть зависимости между работами**, например, нельзя жарить хлеб, пока мы его не нарезали.  
С учетом зависимостей мы получим следующую диаграмму расписания нашего проекта (Рисунок 7.4)

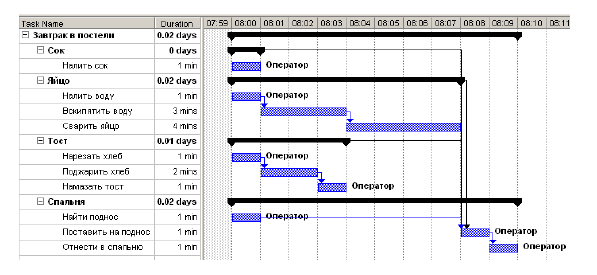
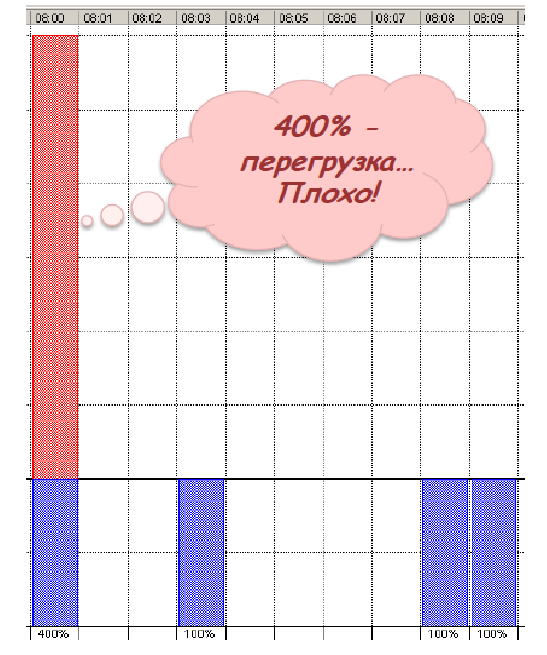
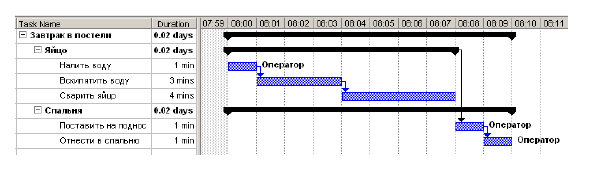


Рисунок 7.4. Диаграмма расписания «суперпроекта» с учетом зависимостей между работами.

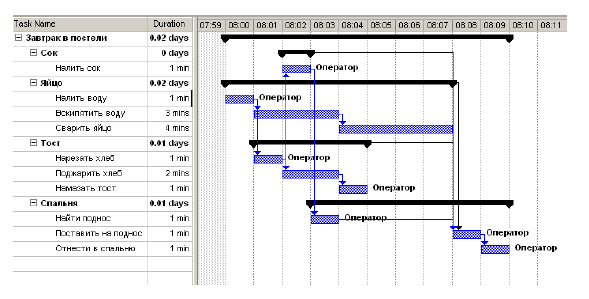
В результате мы определили, что минимальный срок реализации нашего проекта составляет 10 минут. Однако мы не можем на этом остановиться, поскольку должны еще учесть ограничение по ресурсам. У нас только один оператор. Если мы посмотрим **на диаграмму загруженности ресурсов (Рисунок 7.5), то увидим, что наш критический ресурс загружен на первой минуте на 400%. что недопустимо.**

  
Рисунок 7.5. Диаграмма загруженности ресурсов в «суперпроекте»

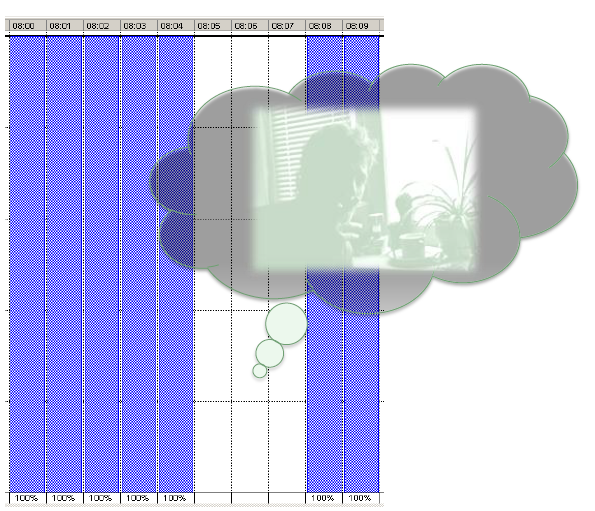
Следовательно, мы должны выполнить выравнивание ресурсов. Поскольку одним из критериев успеха проекта является его минимальная длительность, то если мы не хотим ее увеличивать, мы **должны выявить критический путь в проекте** (Рисунок 7.6) и не сдвигать работы, которые на нем находятся.

  
Рисунок 7.6. Критический путь в «суперпроекте»

Поэтому, после выравнивания ресурсов, расписание нашего проекта будет выглядеть следующим образом (Рисунок 7.7).

  
Рисунок 7.7. Расписание «суперпроекта» после выравнивания ресурсов

Теперь диаграмма загруженности ресурсов (Рисунок 7.8) выглядит приемлемо и у оператора даже появилось три минуты свободного времени на перекур. При этом общая длительность реализации проекта по-прежнему составляет 10 минут.

  
   
Рисунок 7.8. Диаграмма загруженности ресурсов после выравнивания

***Основная литература***

1. Архипенков С. Лекции по управлению программными проектами// Электронный ресурс:http://citforum.ru/SE/project/

2. Основы управления ИТ-проектами [Электронный ресурс]: учебное пособие / сост. Е.Р. Кирколуп, Ю.Г., Скурыдин, Е.М. Скурыдина. – Барнаул : АлтГПУ, 2017// Электронный ресурс:http://library.altspu.ru/dc/pdf/kirkolup.pdf

3. Рыбалова Е.А. Управление проектами: учебное пособие / Е.А. Рыбалова. - Томск: ФДО, ТУСУР, 2015. - 205 с.

4. Бухарев Н.Р. Введение в гибкое управление проектами. Краткий конспект лекций// tulpar.kfu.ru/course/view.php?id=2284

[5. Коптелов А., Никитин И., Цулая М. Гибкая процессная методология Agile](https://www.intuit.ru/studies/courses/3590/832/info). - М.: ИНТУИТ // Электронный ресурс: // www.intuit.ru/studies/courses/3590/832/print\_lecture/31170

6. Топ-7 методов управления проектами: Agile, Scrum, Kanban, PRINCE2 и другие// www.pmservices.ru/project-management-news/top-7-metodov-upravleniya-proektami-agile-scrum-kanban-prince2-i-drugie/

### 7.[Грекул](https://www.intuit.ru/intuituser/userpage/v_i_grekul) В., [Коровкина](https://www.intuit.ru/intuituser/userpage/n_l_korovkina) Н., [Куприянов](https://www.intuit.ru/intuituser/userpage/y_v_kupriyanov) Ю. [Методические основы управления ИТ-проектами](https://www.intuit.ru/studies/courses/646/502/info). - М.: ИНТУИТ // Электронный ресурс: https://www.intuit.ru/studies/courses/646/502/info

### ***Дополнительная литература:***

8. Сериков А. Основы управления проектами. - М.: ИНТУИТ. Электронный ресурс: www.intuit.ru/studies/professional\_retraining/965/courses/272/info

9. Управление проектами: Учебное пособие / М.В. Романова. - М.: ИД ФОРУМ: НИЦ ИНФРА-М, 2014

10. Руководство к Своду знаний по управлению проектами РМВОК (A Guide to the Project Management Body of Knowledge PMBOK®) — 4-е изд. — Project Management Institute, Inc., 2008. — 464 с.

11. https://products.office.com/

**Лекция 8. Управление человеческими ресурсами проекта**

*План:*

1. Команда проекта

2. Организация проектной команды

3. Лидерство и управление

4. Стратегии руководства.

5. Правильные люди

6. Мотивация

7. Эффективное взаимодействие

**1. Команда проекта**

При распределении ролей и ответственности, необходимых для выполнения проекта, следует учитывать следующие моменты.

**Роль в проекте (проектная роль)** – определенный набор функций и полномочий в проекте, созданный с целью распределения обязанностей между членами команды проекта. Проектную роль можно рассматривать как временную должность в организации (компании).

**Полномочия** – право задействовать ресурсы проекта, принимать решения и утверждать одобрение действий или результатов. Примеры полномочий: выбор способа завершения операции, приемка качества и порядок реагирования на отклонения в проекте.

**Ответственность** – работа, которую член команды проекта должен выполнить для завершения операций проекта.

**Квалификация** – навыки и способности, необходимые для выполнения операций проекта. Отсутствие нужной квалификации у членов команды влияет на расписание проекта, качество выполнения работ, ставит под угрозу цели проекта. Для повышения квалификации планируют проведение обучения членов команды.

Формируя команду управления проектом, необходимо определить ключевых лиц проекта, принимающих решения. **Со стороны заказчика ключевыми ролями** являются – спонсор проекта и менеджер проекта со стороны заказчика.

***Спонсор проекта*** обеспечивает организационную сторону проекта и подтверждает правильность целей проекта. В его ведении находится бюджет проекта. Спонсором проекта может быть отдельный человек или целый комитет, в зависимости от масштабов и сложности проекта.

***Менеджер проекта*** со стороны заказчика назначается и в том случае, если осуществление проекта организацией заказчика требует ежедневного управления. В его обязанности входит предоставление ресурсов заказчиков, разрешение проблем и отслеживание состояния проекта.

**Ключевые роли со стороны исполнителя** – руководитель проекта (менеджер проекта) со стороны исполнителя и бизнес-менеджер.

***Бизнес-менеджер*** отвечает за успешное выполнение проекта и представляет исполнителя в его договорных отношениях с заказчиком.

***Менеджер проекта*** (руководитель проекта) отвечает как за успехи, так и за неудачи проекта. В его задачи входит управление сроками, стоимостью, качеством работ с целью удовлетворения ожиданий заказчика и достижения бизнес-целей исполнителя.

**Команда управления проектом** включает ***координатора проекта, администратора проекта, менеджера по конфигурации.*** Для крупных проектов к выполнению каждой из этих ролей (функций) могут быть привлечено нескольких человек. На небольших проектах менеджер проекта может совмещать несколько ролей (функций). Масштабные проекты предполагают наличие менеджера по качеству, который ответственен перед бизнес-менеджером исполнителя. В крупных проектах могут быть организованы комитет по управлению, комитет по контролю за изменениями, комитет по анализу спорных вопросов.

В состав команды проекта входят не только команда управления проектом, но и **исполнители проекта**.

Примеры проектных **ролей исполнителей**, характерных для ИТ-проектов: функциональный архитектор, функциональный консультант, разработчик, администратор ИС, тестировщик, менеджер по качеству, системный аналитик.

### **2. Организация проектной команды**

Каждый проект разработки ПО имеет свою организационную структуру, которая определяет распределение ответственности и полномочий среди участников проекта, а также обязанностей и отношений отчетности. Чем меньше проект, тем больше ролей приходится совмещать одному исполнителю.

***Роли и ответственности участников типового проекта разработки ПО***можно условно разделить на пять групп:

* **Анализ.** Извлечение, документирование и сопровождение требований к продукту.
* **Управление.**Определение и управление производственными процессами.
* **Производство.**Проектирование и разработка ПО.
* **Тестирование.** Тестирование ПО.
* **Обеспечение.**Производство дополнительных продуктов и услуг.

**Группа анализа**включает в себя следующие роли:

* Бизнес-аналитик. Построение модели предметной области (онтологии).
* Бизнес-архитектор. Разрабатывает бизнес-концепцию системы. Определяет общее видение продукта, его интерфейсы, поведение и ограничения.
* Системный аналитик. Отвечает за перевод требований к продукту в функциональные требования к ПО.
* Специалист по требованиям. Документирование и сопровождение требований к продукту.
* Менеджер продукта (функциональный заказчик). Представляет в проекте интересы пользователей продукта.

**Группа управления** состоит из следующих ролей:

* Руководитель проекта. Отвечает за достижение целей проекта при заданных ограничениях (по срокам, бюджету и содержанию), осуществляет операционное управление проектом и выделенными ресурсами.
* Куратор проекта. Оценка планов и исполнения проекта. Выделение ресурсов.
* Системный архитектор. Разработка технической концепции системы. Принятие ключевых проектных решений относительно внутреннего устройства программной системы и её технических интерфейсов.
* Руководитель группы тестирования. Определение целей и стратегии тестирования, управление тестированием.
* Ответственный за управление изменениями, конфигурациями, за сборку и поставку программного продукта.

В **производственную группу** входят:

* Проектировщик. Проектирование компонентов и подсистем в соответствие с общей архитектурой, разработка архитектурно значимых модулей.
* Проектировщик базы данных.
* Проектировщик интерфейса пользователя.
* Разработчик. Проектирование, реализация и отладка отдельных модулей системы.

В большом проекте может быть несколько производственных групп, ответственных за отдельные подсистемы. Как правило, проектировщик выполняет роль лидера группы и управляет своим подпроектом или пакетом работ. Стоит не забывать, что руководитель проекта делегирует полномочия, но не ответственность.

**Группа тестирования** в проекте состоит из следующих ролей:

* Проектировщик тестов. Разработка тестовых сценариев.
* Разработчик автоматизированных тестов.
* Тестировщик. Тестирование продукта. Анализ и документирование результатов.

**Группа обеспечения.** Участники группы обеспечения, как правило, не входят в команду проекта. Они выполняют работы в рамках своей процессной деятельности. К группе обеспечения можно отнести следующие проектные роли:

* Технический писатель.
* Переводчик.
* Дизайнер графического интерфейса.
* Разработчик учебных курсов, тренер.
* Участник рецензирования.
* Продажи и маркетинг.
* Системный администратор.
* Технолог.
* Специалист по инструментальным средствам.
* Другие.

В зависимости от масштаба проекта одну роль могут исполнять несколько человек. Например, разработчики, тестировщики, технические писатели. Некоторые роли всегда должен исполнять только один человек. Например, Руководитель проекта, Системный архитектор. Один человек может исполнять несколько ролей. Возможны следующие совмещения ролей:

* Руководитель проекта + системный аналитик (+ системный архитектор)
* Системный архитектор + разработчик
* Системный аналитик + проектировщик тестов (+ технический писатель)
* Системный аналитик + проектировщик интерфейса пользователя
* Ответственный за управление конфигурациями + ответственный за сборку и поставку (+ разработчик)

Крайне нежелательно совмещать следующие роли:

* Разработчик + руководитель проекта
* Разработчик + системный аналитик.
* Разработчик + проектировщик интерфейсов пользователя.
* Разработчик + тестировщик

Важно помнить, что организационная структура проекта — «живой» организм. Она начинает складываться на стадии планирования и может меняться в ходе проекта. Нестабильность организационной структуры (частые замены исполнителей) — серьезная проблема в управлении сложными программными проектами, поскольку существует время вхождения в контекст проекта, которое может измеряться месяцами.

### **3. Лидерство и управление**

В работе руководителя проекта есть две стороны: управление и лидерство, которые одинаково важны и не могут существовать в отрыве друг от друга. Нельзя быть лидером материальных ресурсов, денежных потоков, планов, графиков и рисков. Ими необходимо управлять. Эффективные команды не образуются сами по себе, они кристаллизуются вокруг признанного лидера. Как не бывает лидеров без последователей, так и не бывает команд без лидеров. Поэтому первый шаг руководителя при создании эффективной команды — это стать лидером, вокруг которого сможет сплотиться рабочий коллектив. Лидера нельзя назначить.

***Лидерство - это умение управлять своей собственной жизнью и только потом другими людьми.***Сверхвысокое значение коэффициента интеллекта IQ, к сожалению, в этом не поможет.

Личная эффективность человека на 80% определяется его ***коэффициентом эмоционального интеллекта*EQ** (Emotional Intelligence)  — способностью понимать и эффективно взаимодействовать с другими людьми. Хорошая новость. В отличие от IQ, который формируется в ранней молодости и затем практически не меняется, EQ можно повышать на протяжении всей жизни. Если, конечно, прилагать к этому усилия.

**Лидер должен получить признание команды.**Для того этого необходимо:

* Признание командой профессиональной компетентности и превосходства лидера.
* Полное доверие команды к действиям и решениям лидера, признание его человеческих качеств, убежденность в его честности, порядочности, вера в его искренность и добросовестность.

**Если руководитель не смог стать лидером, он будет вынужден применять в своей практике управленческие антипаттерны**. Для такого руководителя характерны чрезмерная настороженность, скрытность, неспособность делегировать полномочия. Он исходит из предпосылок индустриальной эпохи Генри Форда: «Работники ленивы, поэтому им необходимы внешние стимулы для работы. У людей нет честолюбия, и они стараются избавиться от ответственности. Чтобы заставить людей трудиться, необходимо использовать принуждение, контроль и угрозу наказания». Манфред Кетс де Врис называет данное отклонение «параноидальным управлением».

Бесполезно пытаться мотивировать участников команды на успех проекта, не исключив из своего руководящего арсенала практики демотивации — антипаттерны, применение которых в управлении творческими коллективами не приносит ничего, кроме вреда. Вместо мотивирования сотрудников на успех, применение антипаттернов мотивирует их на избежание риска и негативных для себя последствий, подавляет свободу, самостоятельность, творчество и инициативу. Это приводит к деструктивному подчинению, когда все работают строго по инструкции и только в соответствие с указаниями руководства, и полному отсутствию личной ответственности исполнителей, «А какие ко мне претензии? Как сказали, так я и сделал!» В результате — низкая эффективность и качество работы, ухудшение морального климата. Вместо доверия и сотрудничества в коллективе процветают подозрительность и формальное взаимодействие. А вы никогда не видели, как тимлид беседует с программистом только «под протокол» и с подписями на каждом листе? Наконец, применение антипаттернов — это стрессы, усталость участников, личные проблемы, увольнение наиболее профессиональных сотрудников и провал проекта.

***Эффективный лидер обязан обладать следующими компетенциями:***

* Видение целей и стратегии их достижения.
* Глубокий анализ проблем и поиск новых возможностей
* Нацеленность на успех, стремление получить наилучшие результаты.
* Способность сочувствия, понимания состояния участников команды.
* Искренность и открытость в общении.
* Навыки в разрешении конфликтов.
* Умение создавать творческую атмосферу и положительный микроклимат.
* Терпимость, умение принимать людей какие они есть, принятие их права на собственное мнение и на ошибку.
* Умение мотивировать правильное профессиональное поведение членов команды.
* Стремление выявлять и реализовывать индивидуальные возможности для профессионального роста каждого.
* Способность активно "обеспечивать", "доставать", "выбивать" и т.д.

### **4. Стратегии руководства.**

В зависимости от готовности участников рабочей группы выполнять задания руководителя, он должен использовать одну из 4-х стратегий:

* ***«Директивное управление».*** Руководитель говорит, указывает, направляет, устанавливает. Жесткое назначение работ, строгий контроль сроков и результатов.
* ***«Объяснения».*** Лидер "продает", объясняет, проясняет, убеждает. Сочетание директивного и коллективного управления. Объяснение своих решений.
* ***«Участие».*** Лидер участвует, поощряет, сотрудничает, проявляет преданность. Приоритетное коллективное принятие решений, обмен идеями, поддержка инициативы подчиненных.
* ***«Делегирование».***Лидер делегирует, наблюдает, обслуживает. «Не мешать» — пассивное управление сформировавшегося лидера.

Применение этих стратегий можно проиллюстрировать на примерах (Рисунок 8.1).

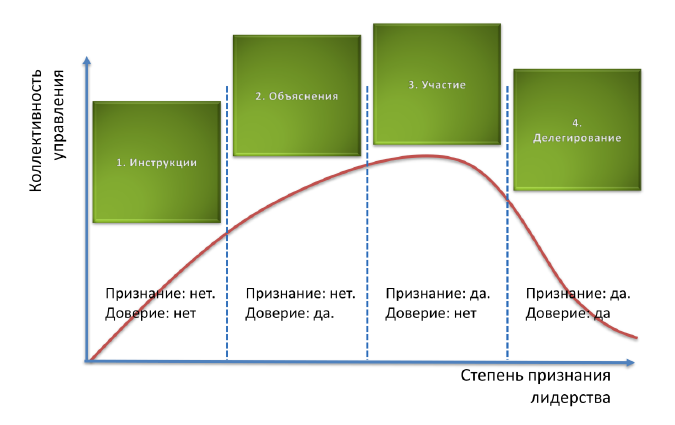


Рисунок 8.1. Ситуационное лидерство

* Вас назначили руководителем в новый коллектив. Вы еще не получили признания, а дело делать надо. ***Стратегия: «Директивное управление».***
* Вы были участником команды. Вас назначили руководителем этой команды. Доверие есть, а уверенности в правильности ваших действий нет. ***Стратегия: «Объяснения».***
* Вас назначили руководителем в новый коллектив. Все знают о ваших прежних сложных и успешных проектах. Все признают ваше превосходство, но доверия к вам нет. Никто не знает, какой ценой были достигнуты ваши победы. ***Стратегия: «Участие».***
* Между вами и участниками установлено взаимное доверие. Все достаточно мотивированы на успех проекта. Каждый сам себе может быть руководителем. ***Стратегия: «Делегирование».***

Основные усилия руководителя, если он стремится получить наивысшую производительность рабочей группы, должны быть направлены на изучение и изменение объекта управления: людей и их взаимодействия. Следовательно, **задачу адаптивного управления мы можем разделить на две подзадачи**:

* Обеспечить эффективность каждого участника рабочей группы.
* Обеспечить эффективные процессы взаимодействия.

 5. **Правильные люди**

Рональд Рейган говорил: «Окружите себя самыми лучшими людьми, которых вы только сможете найти, передайте им в руки власть и не мешайте им».  
***Эффективный командный игрок:***

* Занимает активную позицию, стремится расширить свою ответственность и увеличить личный вклад в общее дело.
* Постоянно приобретет новые профессиональные знания и опыт, выдвигает новые идеи, направленные на повышение эффективности достижения общих целей, добивается распространения своих знаний, опыта и идей среди коллег.
* Получает удовольствие от своей работы, гордится ее результатами и стремится, чтобы эти же чувства испытывали все коллеги.
* Четко осознает свои личные и общие цели, понимает их взаимообусловленность, настойчиво стремится к их достижению.
* Уверен в себе и в своих коллегах, объективно оценивает их достижения и успехи, внимательно относится к их интересам и мнениям, активно ищет взаимовыгодное решение в конфликтах.
* Является оптимистом, при этом твердо знает, что окружающий мир несовершенен; воспринимает каждую новую проблему, как дополнительную возможность подтвердить собственный профессионализм в своих глазах и во мнении коллег.

Вместе с тем, встречаются патологии поведения, которые неприемлемы в команде:

* Непорядочность. Лживость, отсутствие совести и чувства справедливости, способность на низкие поступки.
* Синдром острого дефицита эмпатии. Эгоцентризм. Неуважение и невнимание к партнерам. Склонность к отрицательным оценкам других. Грубость. «Каждый сам за себя! — никто тебе не поможет!» «Человек человеку волк!»
* «Звезданутость». Завышенная самооценка. Подчеркивание собственного превосходства. Умничанье. Человек сильно переоценивает свой личный вклад в общее дело и поэтому считает, что он должен работать меньше, чем его «менее способные» коллеги.
* Вульгарный анархизм. Вольница — это полная безответственность, свобода от каких либо обязательств перед другими, ничем не сдерживаемые проявления чувств, действия или поступки. «Произвольничать, поступать самовольно, в обиду другим, нагло, дерзко» (с) В.Даль. Не путать со «свободой»!
* «Социальный паразитизм». Стремление прожить вольготно за чужой счет там, где ответственность размыта, а личный вклад трудно четко выделить.

Существуют четыре необходимых и достаточных ***условия для того, чтобы сотрудник эффективно решил поставленную задачу***. Это:

* Понимание целей работы.
* Умение ее делать.
* Возможность ее сделать.
* Желание ее сделать.

Для того чтобы обеспечить выполнение этих условий, ***руководитель должен уметь эффективно выполнять четыре функции:***

* **Направлять.** Если сотрудник не понимает что делать, задача руководителя — обеспечить общее видение целей и стратегии их достижения.
* **Обучать.** Если сотрудник не умеет, задача руководителя — «обучать», быть наставником и образцом для подражания.
* **Помогать.** Если у сотрудника не может выполнить работу, задача руководителя — «помогать», обеспечить исполнителя всем необходимым, убрать препятствия с его пути.
* **Вдохновлять.**Если у сотрудника не достаточно желание выполнить работу, задача руководителя — «вдохновить», обеспечить адекватную мотивацию участника на протяжении всего проекта.

6. **Мотивация**

Известно, что ни одна задача не будет решена за любое, отведенное на это время, если человек не захочет ее сделать. Он всегда найдет для оправдания этого 100 «объективных» причин, вместо того, чтобы найти хотя бы один способ решения задачи.

**У каждого участника рабочей группы должна быть личная цель (внутренняя мотивация), которую он сможет достичь, продвигая проект к успеху.**Начните с себя! Вам нужно четко понимать, в чем состоит ваш выигрыш в случае успешного завершения проекта. Добиться от участников приверженности проекту больше, чем имеете вы сами, вам не удастся. Если у участника нет такой значимой личной цели, избавьтесь от него. Иначе вам придется потратить все свое время на «промывание его мозгов» и попытки мотивировать его на эффективную работу.

***Мотивация должна начинаться с подбора сотрудников в команду.***В старой экономике людей нанимали за умения и обучали нужному отношению к делу. В новой экономике необходимо поступать с точностью до наоборот: нанимать за нужное отношение к делу и учить необходимым умениям.

***Люди не рождаются победителями, они ими становятся.*** Кандидата стоит нанимать только в случае, если вы можете предложить ему возможность стать победителем. Настоящий лидер предлагает не работу, а возможности.

Все люди разные, а ситуаций, в которых они могут находиться в ходе проекта, бесчисленное множество. Бойтесь стереотипов. Если вы не учитываете индивидуальные особенности конкретной личности, то эффективность ваших взаимодействий сильно снижается.

Модель объекта управления нам неизвестна, следовательно, не может существовать исчерпывающий набор правил, типа «если..., то...», по которым смог бы действовать руководитель. Поэтому, сколько людей и ситуаций, столько и вариантов решений должен иметь эффективный руководитель в своем запасе. «Если у руководителя в руках только молоток, то все вокруг будут похожи на гвозди».

***Руководитель при поиске решения опирается на свой багаж знаний и умений.*** Он пытается понять каждого участника, классифицировать состояние, найти в своем опыте похожую ситуацию и адаптировать ранее использованное успешное решение применительно к данному конкретному случаю. Таким образом, руководитель стремится помочь человеку (объекту управления) перейти в новое более эффективное с точки зрения целей проекта состояние.

Затем***руководитель должен наблюдать за результатами своего воздействия — это и есть дополнительный контур обратной связи.***Необходимо помнить, что понять человека можно, только слушая и слыша, что он говорит. Руководитель, который в течение недели не пообщался индивидуально с каждым из своих прямых подчиненных, зря получает зарплату. И совсем не обязательно разговор должен идти о статусе проектных работ. Порой, достаточно поговорить о погоде, кино или футболе. После этого руководитель анализирует полученные результаты и аккумулирует новый опыт (положительный или отрицательный) в своей «базе знаний».

***Чем опытней руководитель, тем точнее он может распознать и классифицировать сложившуюся ситуацию***, тем больше в его «базе знаний» прецедентов, используя которые, он может синтезировать решение для данного конкретного случая. Именно поэтому в управлении программными проектами в первую очередь ценится опыт руководителя и только потом, возможно, его звания и знания.

Программист состоит из четырех компонентов: тело, сердце, разум и душа.

* Телу необходимы деньги и безопасность.
* Сердцу — любовь и признание.
* Разуму — развитие и самосовершенствование.
* Душе — самореализация

Предоставьте все это вашим сотрудникам, и эффективность их труда возрастет многократно. В противном случае люди, которые хотят побеждать, найдут все это в другой команде, а в вашей останутся только неудачники.

### **7. Эффективное взаимодействие**

***Процесс производства программного обеспечения, применяемый в проекте, должен основываться на итеративности, инкрементальности, самоуправляемости команды и адаптивности.***

**Главный принцип:** **не люди должны строиться под выбранную модель процесса, а модель процесса должна подстраиваться под конкретную команду,** чтобы обеспечить ее наивысшую производительность.

В программной инженерии многие уже признали, что наиболее эффективные производственные процессы складываются в самоуправляемых и самоорганизующихся рабочих командах. Идеи командного менеджмента на Западе зародилась в начале 80-х годов. Эффективность команд в новых экономических условиях одними из первых оценили такие гиганты, как Procter & Gamble и Boeing. Доктрина командного менеджмента предполагает ясность общих ценностей и целей, самоорганизацию и самоуправление совместной деятельностью, взаимный контроль, взаимопомощь и взаимозаменяемость, коллективную ответственность за результаты труда, всемерное развитие и использование индивидуального и группового потенциалов.

Если на Западе идеи командного менеджмента только зарождаются, то для России они имеют давние традиции. С XIII века в России существовали ***производственные артели*** — различные формы добровольных объединений людей с целью осуществления общей хозяйственной деятельности. Артель была добровольным товариществом совершенно равноправных работников, призванных на основе взаимопомощи и взаимовыручки решать практически любые хозяйственные и производственные задачи. Известный российский писатель и революционер, А.И. Герцен видел в коллективизме, в уникальности существующей сельской общины, в городской артели и в самобытной воинской организации казачества характерную черту жизни и психологии русского народа. Позднее во времена СССР широкое распространение на производстве получили **рабочие бригады**, а в научно-исследовательских институтах — временные **научные творческие коллективы**, которые объединялись на основе общности цели, взаимопомощи и коллективной ответственности за результат.  
Истинный российский коллективизм (соборность) не имеет ничего общего с вульгарным коллективизмом: со стадностью, подавлением личности («я нет, есть только мы!», «я — последняя буква алфавита!»), уступчивостью, групповой психологией и слепым подчинением меньшинства большинству. Суть российского коллективизма в том, что человек ощущает себя элементом органичной системы, где он выполняет свою функцию, свою задачу, совершенно особенную, которую не выполняет никто другой и выполнить не может. Эту задачу он выполняет совершенно сознательно, стремясь к максимальной реализации своих целей.

***Руководителю недостаточно стать лидером, надо еще суметь сплотить команду.***Эксперты в области командного менеджмента выделяют ***4 обязательные последовательные стадии***, через которые должна пройти рабочая группа прежде, чем она станет эффективной командой.

* ***Forming. Формирование.*** Характеризуется избытком энтузиазма, связанного с новизной. Люди должны преодолеть внутренние противоречия, переболеть конфликтами прежде, чем сформируется действительно спаянный коллектив. На этом этапе многое зависит от руководителя. Он должен четко поставить цели членам команды, верно определить роль каждого в проекте.
* **Storming.*Разногласия и конфликты.***Самый сложный и опасный период. Мотивация новизны уже исчезла, а сильные и глубокие стимулы у команды еще не появились. Неизбежные сложности или неудачи порождают конфликты и «поиск виновных». Участники команды методом проб и ошибок вырабатывают наиболее эффективные процессы взаимодействия. Руководителю на этом этапе важно обеспечить открытую коммуникацию в команде. Конфликты не следует прятать или разрубать. Споры необходимо разруливать спокойно, терпеливо и тщательно.
* **Norming.*Становление.***В команде растет доверие, люди начинают замечать в коллегах не только проблемные, но и сильные стороны. Закрепляются и оттачиваются наиболее эффективные процессы взаимодействия. На смену битве амбиций приходит продуктивное сотрудничество. Четче становится разделение труда, исчезает дублирование функций. Руководитель перестает находиться в состоянии постоянного аврала, работа по построению команды на этом этапе — уже не тушение пожара, а скрупулезный труд по отработке общих норм и правил.
* **Performing.*Отдача.*** Команда работает эффективно, высок командный дух, люди хорошо знают друг друга и умеют использовать сильные стороны коллег. Все стремятся придерживаться выработанных общих процессов. Высок уровень доверия. Это лучший период для раскрытия индивидуальных талантов.

Часто случается, что рабочая группа вязнет на одной из стадий и никогда не достигает плато наивысшей производительности. Если команда прошла все стадии формирования и вышла на фазу «Performing», не стоит полагать, что менеджер проекта может делегировать все свои полномочия и отправиться в отпуск. Задача менеджера на этом этапе — «точить пилу». Это значит поддерживать требуемый уровень мотивации, быть штурманом, искать новые пути и открывать новые возможности. Постоянно наблюдать и оценивать эффективность всех процессов, применяемых в проекте. Искать ответ на вопросы: «Что угрожает проекту?» «Что лишнее мы делаем?» «Что можно делать проще?» Работать на сокращение ненужных усилий вместо того, чтобы «стремиться к новым героическим подвигам».

Если руководитель не будет прилагать дополнительные усилия команда, рано или поздно, начнет «сползать» с плато наивысшей эффективности в состояние застоя и стагнации (Рисунок 8.2).

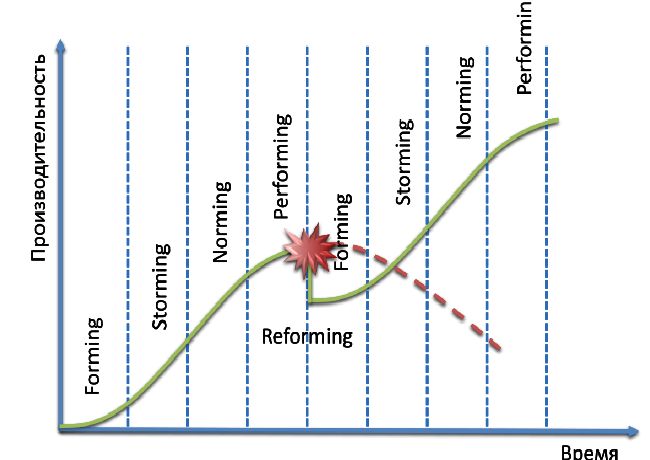


Рисунок 8.2. Reforming. «Встряхивание» и перевод команды проекта на новый, более высокий, уровень производительности.

Помните, что окружение и команда изменяются по ходу проекта. Прежняя мотивация ослабевает или перестает действовать. Изменяйте правила и процессы. Отказывайтесь от того, что перестало действовать или стало работать неэффективно. «Встряхивайте» (Reforming) и возвращайте команду в стадию «Forming». Это позволит ей снова, пройдя через все этапы становления, выйти на новый более высокий уровень производительности. Разумеется, делать это следует, после сдачи очередного релиза программного продукта, ну и, возможно, в случае глубокого кризиса проекта.

### 

### Выводы

Эффективные команды не образуются сами по себе, они кристаллизуются вокруг признанного лидера. Для того чтобы стать лидером, необходимо:

* Признание командой профессиональной компетентности и превосходства лидера.
* Полное доверие команды к действиям и решениям лидера, признание его человеческих качеств, убежденность в его честности, порядочности, вера в его искренность и добросовестность.

Мотивация должна начинаться с подбора сотрудников в команду. В старой экономике людей нанимали за умения и обучали нужному отношению к делу. В новой экономике необходимо поступать с точностью до наоборот: нанимать за нужное отношение к делу и учить необходимым умениям.

Рабочая группа прежде, чем она станет эффективной командой, должна пройти четыре обязательных последовательных стадии:

1) Forming,

2) Storming,

3) Norming,

4) Performing.

Если руководитель не будет прилагать дополнительные усилия команда, рано или поздно, начнет «сползать» с плато наивысшей эффективности в состояние застоя и стагнации. Задача менеджера на этом этапе — поддерживать требуемый уровень мотивации, быть штурманом, искать новые пути и открывать новые возможности.

Четыре стадии развития команды должны циклически повторяться, чтобы обеспечить непрерывный рост производительности.

***Основная литература***

1. Архипенков С. Лекции по управлению программными проектами// Электронный ресурс:http://citforum.ru/SE/project/

2. Основы управления ИТ-проектами [Электронный ресурс]: учебное пособие / сост. Е.Р. Кирколуп, Ю.Г., Скурыдин, Е.М. Скурыдина. – Барнаул : АлтГПУ, 2017// Электронный ресурс:http://library.altspu.ru/dc/pdf/kirkolup.pdf

3. Рыбалова Е.А. Управление проектами: учебное пособие / Е.А. Рыбалова. - Томск: ФДО, ТУСУР, 2015. - 205 с.

4. Богданов В.В. Управление проектами в Microsoft Project 2007. Учебный курс. -  СПб: Питер, 2010

[5. Коптелов А., Никитин И., Цулая М. Гибкая процессная методология Agile](https://www.intuit.ru/studies/courses/3590/832/info). - М.: ИНТУИТ // Электронный ресурс: // www.intuit.ru/studies/courses/3590/832/print\_lecture/31170

6. Топ-7 методов управления проектами: Agile, Scrum, Kanban, PRINCE2 и другие// www.pmservices.ru/project-management-news/top-7-metodov-upravleniya-proektami-agile-scrum-kanban-prince2-i-drugie/

### 7. [Грекул](https://www.intuit.ru/intuituser/userpage/v_i_grekul) В., [Коровкина](https://www.intuit.ru/intuituser/userpage/n_l_korovkina) Н., [Куприянов](https://www.intuit.ru/intuituser/userpage/y_v_kupriyanov) Ю. [Методические основы управления ИТ-проектами](https://www.intuit.ru/studies/courses/646/502/info). - М.: ИНТУИТ // Электронный ресурс: https://www.intuit.ru/studies/courses/646/502/info

### ***Дополнительная литература:***

8. Сериков А. Основы управления проектами. - М.: ИНТУИТ. Электронный ресурс: www.intuit.ru/studies/professional\_retraining/965/courses/272/info

9. Управление проектами: Учебное пособие / М.В. Романова. - М.: ИД ФОРУМ: НИЦ ИНФРА-М, 2014

10. Руководство к Своду знаний по управлению проектами РМВОК (A Guide to the Project Management Body of Knowledge PMBOK®) — 4-е изд. — Project Management Institute, Inc., 2008. — 464 с.

11. https://products.office.com/

### 12. Архипенков С. [Руководство командой разработчиков программного обеспечения. Прикладные мысли. -](http://citforum.ru/SE/project/app_thoughts/)2008

13. Стивен Р. Кови. 7 навыков высокоэффективных людей. Мощные инструменты развития личности, 2-е изд. - М., Альпина Бизнес Букс, 2007.

14. Кетс де Врис М. Мистика лидерства. Развитие эмоционального интеллекта. - М.: Альпина Бизнес Букс, 2005.

15. Hersey P., Blanchard K.H. Management of Organizational Behavio. - Englewood Cliffs: Prentice-Hall, 1993.

16. Томпсон Л. Создание команды. -М.: Вершина, 2005.

**Лекция 9. Управление стоимостью проекта**

*План:*

1. Управление стоимостью проекта

2. Оценка стоимости

3. Инструменты и методы оценки стоимости

4. Определение бюджета

**1. Управление стоимостью проекта**

***Управление стоимостью***  *проекта объединяет процессы, выполняемые в ходе планирования, разработки бюджета и управления расходами, обеспечивающие завершение проекта в рамках утвержденного бюджета.*

Общая блок-схема *процессов управления стоимостью* проекта:

1. *Оценка стоимости* — процесс определения примерной стоимости ресурсов, необходимых для выполнения операций проекта.

2. *Определение бюджета* — процесс суммирования оценок стоимости отдельных операций или пакетов работ для формирования санкционированного базового плана по стоимости.

3. *Управление стоимостью* — процесс мониторинга статуса проекта для корректировки бюджета проекта и внесения изменений в базовый план по стоимости.

Данные процессы взаимосвязаны друг с другом, а также с процессами других областей знаний. Каждый процесс происходит в каждом проекте не менее одного раза и выполняется в одной или нескольких фазах проекта. Возможности влияния на стоимость максимальны на ранних стадиях проекта, поэтому очень важно как можно раньше определить содержание.

Процессы управления стоимостью и связанные с ними инструменты и методы обычно выбираются на стадии определения жизненного цикла проекта и документально фиксируются в плане управления стоимостью.

• *Единицы измерения* для каждого типа ресурсов оговариваются (например, человеко-час, человеко-дни, недели, фиксированная стоимость).

• *Связи между процедурами организации* — иерархическая структура работ (ИСР) предоставляет структуру для плана управления стоимостью, что позволяет обеспечить совместимость оценок, бюджета и управления стоимостью. *Элемент ИСР*, используемый для учета стоимости проекта, называется *контрольным счетом*. Каждому контрольному счету присваивается уникальный код, который непосредственно связан с системой бухгалтерского учета исполняющей организации.

• *Контрольные пороги* — для мониторинга выполнения стоимости могут определяться *пороги отклонений*, что позволяет установить заранее согласованную величину допустимого отклонения, прежде чем будут предприняты некоторые действия. Пороги обычно выражаются в отклонении от базового плана, выраженном *в процентах*.

• *Правила измерения исполнения* — устанавливаются правила в соответствии с управлением *освоенным объемом*. Например, план управления стоимостью может определять ИСР и точки, в которых будет проводиться измерение контрольных счетов; устанавливать методы измерения освоенного объема для применения; определять формулы расчета для управления освоенным объемом, необходимые для определения *прогноза по завершении* (ЕАС) и других методов отслеживания.

• *Форматы отчетности* — определяются форматы и регулярность составления разнообразных отчетов о стоимости. Вся эта информация включается в план управления стоимостью (элемент плана управления проектом). Управление стоимостью проекта касается стоимости ресурсов, необходимых для выполнения операций проекта, и охватывает процессы, такие как анализ рентабельности *инвестиций, дисконтированного денежного потока и окупаемости* инвестируемых средств.

Вся эта информация включается в план управления стоимостью (элемент плана управления проектом). Управление стоимостью проекта касается стоимости ресурсов, необходимых для выполнения операций проекта, и охватывает процессы, такие как анализ рентабельности *инвестиций, дисконтированного денежного потока и окупаемости* инвестируемых средств.

**2. Оценка стоимости**

***Оценка стоимости*** *представляет собой процесс разработки приблизительной оценки стоимости ресурсов, необходимых для выполнения операций проекта.*

Оценки стоимости являются *прогнозами*, основанными на информации, известной в конкретный момент времени. Они включают в себя выявление и рассмотрение *альтернатив расчета стоимости* для инициации и выполнения проекта. Для достижения оптимальных затрат проекта должны быть рассмотрены соотношения и риски стоимости, такие как решения «производить или купить», «купить или взять в аренду», а также распределение ресурсов. Точность оценки стоимости проекта повышается по мере продвижения проекта по жизненному циклу (рисунок 9.1).

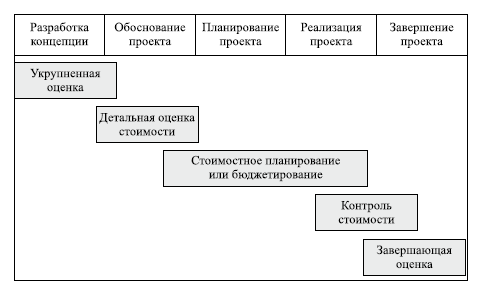


Рисунок 9.1. Управление стоимостью на жизненном цикле проекта

Оценка стоимости является *итеративным* процессом, повторяющимся от фазы к фазе. Например, в фазе *инициации* проекта может быть получена весьма грубая оценка «порядка величины», в диапазоне ±50%. В дальнейшем, по мере поступления информации, диапазон оценки может сузиться до ±10%. Оценивается стоимость всех ресурсов, в частности рабочая сила, материалы, оборудование, услуги и сооружения, особые статьи расходов, такие как учет уровня инфляции, расходы на возможные потери.

Оценка стоимости — это *количественная оценка* возможной стоимости ресурсов, необходимых для выполнения операции

**3. Инструменты и методы оценки стоимости**

1. *Экспертная оценка* — оценки, основанные на исторической информации, дают важное понимание окружающей среды и информации из предыдущих подобных проектов.

2. *Оценка по аналогам* — в оценке стоимости используются значения таких параметров, как *содержание, стоимость, бюджет и длительность* из предыдущих подобных проектов в качестве основы для оценки аналогичных параметров текущего проекта. В качестве основы оценки стоимости текущего проекта принимается *фактическая стоимость* предыдущих подобных проектов. Метод этот обходится дешевле и занимает меньше времени, чем другие методы, но при этом он обычно оказывается и менее точным.

3. *Параметрическая оценка* — это метод, при котором для вычисления оценки параметров операции, таких как стоимость, бюджет и длительность, используются *статистические взаимосвязи* между историческими данными и другими переменными (например, площадью в квадратных метрах в строительстве). При помощи данного метода можно получить более *точную* оценку стоимости.

4. *Оценка «снизу вверх»* представляет собой точный метод оценки элементов работ. Детальная стоимость суммируется или «свертывается» до более высоких уровней с целью последующего отслеживания и составления отчетов.

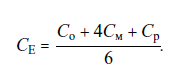
5. *Оценки по трем точкам* происходят из метода **оценки и анализа программ (PERT).** Точность оценок стоимости операций по одной точке может быть улучшена путем рассмотрения *неопределенностей и рисков* оценок. Для определения примерного диапазона стоимости операции метод PERT использует три оценки:

• *Наиболее вероятная* (*C*м) — стоимость операции, основанная на реалистичной оценке трудоемкости работы и всех прогнозируемых расходов.

• *Оптимистическая* (*C*о) — стоимость операции, основанная на анализе благоприятного сценария развития операции.

• *Пессимистическая* (*C*р) — стоимость операции, основанная на анализе неблагоприятного сценария развития операции.

Анализ PERT позволяет определить ***ожидаемую*** (*C*Е) стоимость операции путем вычисления средневзвешенного этих трех оценок:

** (1)

Оценка стоимости, основанная на данном уравнении, может быть более точной, а три точки позволяют прояснить диапазон неопределенности оценки стоимости.

6. *Анализ резервов* — оценки стоимости могут включать в себя резервы на возможные потери для учета неопределенности стоимости. *Резерв на возможные потери* может выражаться в процентах оценочной стоимости, фиксированным числом или может быть разработан с помощью методов количественного анализа. По мере поступления более точной информации о проекте *резервы* на возможные потери являются частью требований к финансированию и могут быть использованы, сокращены или ликвидированы.

7. *Стоимость качества* — для подготовки оценки стоимости операций могут быть использованы допущения о стоимости качества.

8. *Программное обеспечение для управления проектами* может широко использоваться для оценки стоимости проектов. Например, отдельные приложения, электронные таблицы, инструментальные средства моделирования и обработки статистической информации облегчают использование некоторых методов оценки стоимости и способствуют быстрому рассмотрению альтернативных оценок стоимости.

9. *Анализ предложений поставщиков* — методы оценки стоимости могут включать анализ возможной стоимости проекта в зависимости от предложений квалифицированных поставщиков.

**4. Определение бюджета**

***Определение бюджета*** — *процесс объединения оценочных стоимостей отдельных операций или пакетов работ для разработки* ***санкционированного базового*** *плана по стоимости, включающий в себя все бюджеты, за исключением управленческих резервов.*

***Бюджеты проекта*** *представляют собой денежные средства, санкционированные для выполнения проекта. Выполнение стоимости проекта сравнивается с бюджетом.*

***Инструменты и методы определения бюджета***

1. *Суммирование стоимости* — суммирование по пакетам работ в соответствии с ИСР, которые затем объединяются в элементы более высоких уровней элементов ИСР (контрольные счета), в итоге образуется оценка стоимости всего проекта.

2. *Анализ резервов бюджета* может установить как резервы на возможные потери, так и управленческие резервы проекта. *Резервы на возможные потери* представляют собой денежные средства на случай незапланированных, но потенциально необходимых изменений, которые могут возникнуть в результате реализованных рисков, указанных в реестре рисков. *Управленческие резервы* — это бюджеты, зарезервированные на незапланированные изменения содержания и стоимости проекта. *Резервы* не являются частью базового плана проекта по стоимости, но они могут быть включены в общий бюджет проекта и не учитываются при расчете освоенного объема.

3. *Экспертная оценка* — при *определении бюджета* должны использоваться оценки, основанные на опыте в прикладной области, области знаний, сфере деятельности, отрасли промышленности и т. д., в соответствии с выполняемой операцией. Такую экспертную оценку может предоставить группа лиц, обладающих специальным образованием, знаниями, навыками, опытом или подготовкой.

***Управление стоимостью*** *представляет собой процесс* ***мониторинга*** *статуса проекта для корректировки бюджета проекта и внесения изменений в базовый план по стоимости* (рисунок 9.2).

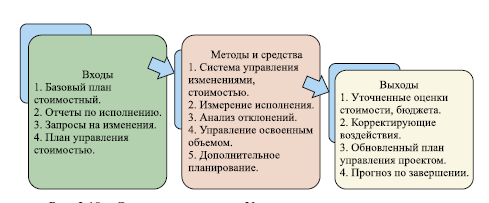


Рисунок 9.2. Структура процесса «Управление стоимостью проекта»

Корректирование бюджета связано с регистрацией фактических затрат, понесенных на определенную дату. Любое увеличение бюджета может быть утверждено только посредством процесса общего *управления изменениями*.

**Управление стоимостью проекта** включает:

• влияние на факторы, вызывающие изменения санкционированного базового плана по стоимости;

• обеспечение своевременной обработки всех запросов на изменение;

• управление фактическими изменениями по мере их возникновения;

• обеспечение расходования средств в рамках утвержденного бюджета в течение определенного периода или на протяжении всего проекта;

• мониторинг выполнения стоимости с целью обнаружения и анализа отклонений от одобренного базового плана по стоимости;

• мониторинг выполнения работ и их сопоставление с затраченными средствами; действия по сокращению ожидаемого перерасхода средств до приемлемого уровня.

*Выводы:*

*Управление стоимостью* проекта включает в себя поиск *причин*, вызывающих как положительные, так и отрицательные отклонения, и является частью процесса осуществления *общего управления изменениями*.

***Основная литература***

1. Архипенков С. Лекции по управлению программными проектами// Электронный ресурс:http://citforum.ru/SE/project/

2. Основы управления ИТ-проектами [Электронный ресурс]: учебное пособие / сост. Е.Р. Кирколуп, Ю.Г., Скурыдин, Е.М. Скурыдина. – Барнаул : АлтГПУ, 2017// Электронный ресурс:http://library.altspu.ru/dc/pdf/kirkolup.pdf

3. Рыбалова Е.А. Управление проектами: учебное пособие / Е.А. Рыбалова. - Томск: ФДО, ТУСУР, 2015. - 205 с.

4. Богданов В.В. Управление проектами в Microsoft Project 2007. Учебный курс. -  СПб: Питер, 2010

[5. Коптелов А., Никитин И., Цулая М. Гибкая процессная методология Agile](https://www.intuit.ru/studies/courses/3590/832/info). - М.: ИНТУИТ // Электронный ресурс: // www.intuit.ru/studies/courses/3590/832/print\_lecture/31170

6. Топ-7 методов управления проектами: Agile, Scrum, Kanban, PRINCE2 и другие// www.pmservices.ru/project-management-news/top-7-metodov-upravleniya-proektami-agile-scrum-kanban-prince2-i-drugie/

### 7. [Грекул](https://www.intuit.ru/intuituser/userpage/v_i_grekul) В., [Коровкина](https://www.intuit.ru/intuituser/userpage/n_l_korovkina) Н., [Куприянов](https://www.intuit.ru/intuituser/userpage/y_v_kupriyanov) Ю. [Методические основы управления ИТ-проектами](https://www.intuit.ru/studies/courses/646/502/info). - М.: ИНТУИТ // Электронный ресурс: https://www.intuit.ru/studies/courses/646/502/info

### ***Дополнительная литература:***

8. Сериков А. Основы управления проектами. - М.: ИНТУИТ. Электронный ресурс: www.intuit.ru/studies/professional\_retraining/965/courses/272/info

9. Управление проектами: Учебное пособие / М.В. Романова. - М.: ИД ФОРУМ: НИЦ ИНФРА-М, 2014

10. Руководство к Своду знаний по управлению проектами РМВОК (A Guide to the Project Management Body of Knowledge PMBOK®) — 4-е изд. — Project Management Institute, Inc., 2008. — 464 с.

11. https://products.office.com/

**Лекция 10. Управление рисками проекта**

*План:*

* 1. Основные понятия

### **2. Планирование управления рисками**

### **3. Идентификация рисков**

### 4. Контрольные списки рисков проектов разработки ПО

### **5. Качественный анализ рисков**

### **6. Количественный анализ рисков**

### **7. Планирование реагирования на риски**

### **8. Главные риски программных проектов и способы реагирования**

### **9. Управление проектом, направленное на снижение рисков**

### **10. Мониторинг и контроль рисков**

**1. Основные понятия**

В силу специфики отрасли, программная инженерия остается и, в ближайшем будущем, будет оставаться производством с высоким уровнем рисков. Управляя проектом разработки ПО, направлено на борьбу с рисками не уложиться в срок, перерасходовать ресурсы, разработать не тот продукт, который требуется.

***Риск это проблема, которая еще не возникла, а проблема — это риск, который материализовался.***

**Риск характеризуется следующими показателями**:

* **Причина или источник.**Явление, обстоятельство обусловливающее наступление риска.
* **Симптомы риска,** указание на то, что событие риска произошло или вот-вот произойдет. Первопричина нам может быть не наблюдаема, например, заразились гриппом. Мы наблюдаем некоторые симптомы — поднялась температура.
* **Последствия риска.** Проблема или возможность, которая может реализоваться в проекте в результате произошедшего риска.
* **Влияние риска.** Влияние реализовавшегося риска на возможность достижения целей проекта. Воздействие обычно касается стоимости, графика и технических характеристик разрабатываемого продукта. Многие риски происходят частично и оказывают соразмерное отрицательное или положительное воздействие на проект.

Примеры характеристик риска показаны на рисунке 10.1.



Рисунок 10.1. Пример характеристик риска

 Риск это всегда вероятность и последствия. Например, всегда есть вероятность того, что метеорит упадет на офис центра программных разработок, и это будет иметь катастрофические последствия для проекта. Однако вероятность наступления этого события настолько мала, что мы в большинстве проектов принимаем это риск и не пытаемся им управлять.  
Майк Ньюэлл, вице-президент компании PSM Consulting, рассказывал, как он объясняет аудитории на своих лекциях, что такое риск.

Он предлагает сыграть в кости на таких условиях, если на кубике выпадает шестерка, то выигрывает он. Если — любое другое число, то выигрывает слушатель. Ставка по 1 доллару. Обычно, большая часть аудитории соглашается сыграть на таких условиях. Майк поднимает ставки: $10, $100, $1000. Постепенно количество желающих поиграть становится все меньше и меньше. При ставке $1000, как правило, желающих рисковать не остается.

Принято выделять ***две категории рисков***:

* «Известные неизвестные». Это те риски, которые можно идентифицировать и подвергнуть анализу. В отношении таких рисков можно спланировать ответные действия.
* «Неизвестные неизвестные». Риски, которые невозможно идентифицировать и, следовательно, невозможно спланировать ответные действия.

***Неизвестные риски это непредвиденные обстоятельства.***Единственное, что мы можем в этом случае предпринять, это создать управленческий резерв бюджета проекта на случай незапланированных, но потенциально возможных изменений.   
Девиз разработчиков ПО из Microsoft: «Мы не боремся с рисками — мы ими управляем».

Цели управления рисками проекта — снижение вероятности возникновения и/или значимости воздействия неблагоприятных для проекта событий. Адекватное управление рисками в компании — признак зрелости производственных процессов.

Отказываться от управления проектными рисками это все равно, что в кинотеатре не иметь огнетушителей и плана эвакуации на случай пожара.

### **2. Планирование управления рисками**

***Управление рисками***это определенная деятельность, которая выполняется в проекте от его начала до завершения. Как и любая другая работа в проекте управление рисками требует времени и затрат ресурсов. Поэтому эта работа обязательно должна планироваться.

***Планирование управления рисками*** — это процесс определения подходов и планирования операций по управлению рисками проекта. Тщательное и подробное планирование управления рисками позволяет:

* выделить достаточное количество времени и ресурсов для выполнения операций по управлению рисками,
* определить общие основания для оценки рисков,
* повысить вероятность успешного достижения результатов проекта.

Планирование управления рисками должен быть завершено на ранней стадии планирования проекта, поскольку оно крайне важно для успешного выполнения других процессов.

Исходными данными для планирования управления рисками служат:

* **Отношение к риску**и толерантность к риску организаций и лиц, участвующих в проекте, оказывает влияние на план управления проектом. Оно должно быть зафиксировано в изложении основных принципов и подходов к управлению рисками.
* **Стандарты организации.** Организации могут иметь заранее разработанные подходы к управлению рисками, например категории рисков, общие определение понятий и терминов, стандартные шаблоны, схемы распределения ролей и ответственности, а также определенные уровни полномочий для принятия решений.
* **Описание содержания проекта**подробно описывает результаты поставки проекта и работы, необходимые для создания этих результатов поставки.
* **План управления проектом**, формальный документ, в котором указано, как будет исполняться проект и как будет происходить мониторинг и управление проектом.

***План управления рисками*** обычно включает в себя следующие элементы:

* **Определение подходов**, инструментов и источников данных, которые могут использоваться для управления рисками в данном проекте.
* **Распределение ролей и ответственности.**Список позиций выполнения, поддержки и управления рисками для каждого вида операций, включенных в план управления рисками, назначение сотрудников на эти позиции и разъяснение их ответственности.
* **Выделение ресурсов и оценка стоимости мероприятий,** необходимых для управления рисками. Эти данные включаются в базовый план по стоимости проекта.
* **Определение сроков**и частоты выполнения процесса управления рисками на протяжении всего жизненного цикла проекта, а также определение операций по управлению рисками, которые необходимо включить в расписание проекта.
* **Категории рисков.**Структура, на основании которой производится систематическая и всесторонняя идентификация рисков с нужной степенью детализации. Такую структуру можно разработать с помощью составления иерархической структуры рисков (Рисунок 10.2).
* **Общие подходы** для определения уровней вероятности, шкалы воздействия и близости рисков на проект.

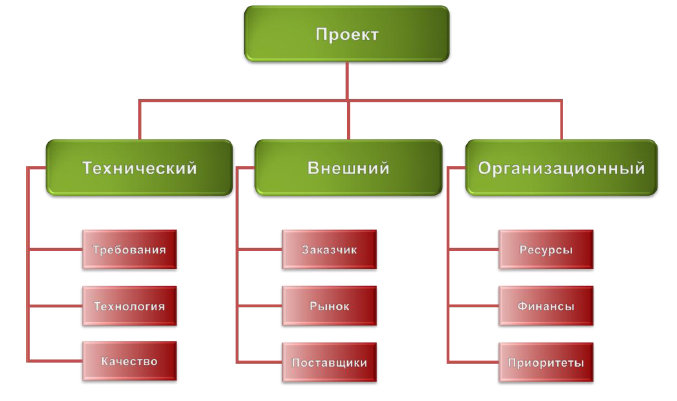


Рисунок 10.2. Пример иерархической структуры рисков проекта

**Шкала оценки воздействия отражает значимость риска** (Таблица 1) в случае его возникновения. Шкала оценки воздействия может различаться в зависимости от потенциально затронутой риском цели, типа и размера проекта, принятыми в организации стратегиями и его финансовым состоянием, а также от чувствительности организации к конкретному виду воздействий.

Таблица 1. Пример шкалы оценки воздействия рисков

|  |  |  |
| --- | --- | --- |
| **Вес** | **Значение** | **Критерий** |
| 3 | Катастрофические | Потери более $100K |
| 2 | Критичные | Потери от $10K до $100K |
| 1 | Умеренные | Потери менее $10K |

Хотя риск может воздействовать и на сроки проекта, и на качество получаемого продукта, но все эти отклонения могут быть оценены в денежном эквиваленте. Например, последствия задержка по срокам для заказной разработки может быть выражена в сумме денежных санкций, определенных в контракте.

Похожая шкала может быть применена для оценки вероятности наступления риска (Таблица 2).

Таблица 2. Пример шкалы оценки вероятности осуществления риска

|  |  |  |
| --- | --- | --- |
| **Вес** | **Значение** | **Критерий** |
| 3 | Очень вероятно | Шансы наступления весьма велики |
| 2 | Возможно | Шансы равны |
| 1 | Маловероятно | Наступление события весьма сомнительно |

Еще одной важной характеристикой риска является близость его наступления. Естественно, что при прочих равных условиях рискам, которые могут осуществиться уже завтра, следует сегодня уделять больше внимания, чем тем, которые могут произойти не ранее, чем через полгода. Для шкалы оценки близости риска может быть применена, например, следующая градация: очень скоро, не очень скоро, очень нескоро.

### **3. Идентификация рисков**

***Идентификация рисков*** — это выявление рисков, способных повлиять на проект, и документальное оформление их характеристик. Это итеративный процесс, который периодически повторяется на всем протяжении проекта, поскольку в рамках его жизненного цикла могут обнаруживаться новые риски.

Исходные данные для выявления и описания характеристик рисков могут браться из разных источников.

В первую очередь это база знаний организации. Информация о выполнении прежних проектов может быть доступна в архивах предыдущих проектов. Следует помнить, что проблемы завершенных и выполняемых проектов, это, как правило, риски в новых проектах.

Другим источником данных о рисках проекта может служить разнообразная информация из открытых источников, научных работ, маркетинговая аналитика и другие исследовательские работы в данной области. Наконец, многие форумы по программированию могут дать бесценную информацию о возникших ранее проблемах в похожих проектах.

**Каждый проект задумывается и разрабатывается на основании ряда гипотез, сценариев и допущений.** Как правило, в описании содержания проекта перечисляются принятые допущения — факторы, которые для целей планирования считаются верными, реальными или определенными без привлечения доказательств. Неопределенность в допущениях проекта следует также обязательно рассматривать в качестве потенциального источника возникновения рисков проекта. Анализ допущения позволяет идентифицировать риски проекта, происходящие от неточности, несовместимости или неполноты допущений.

Для сбора информации о рисках могут применяться различные подходы.

***Основные подходы сбора информации*** о рисках***:***

* Опрос экспертов
* Мозговой штурм
* Метод Дельфи
* Карточки Кроуфорда

***Опрос экспертов.***Цель опроса экспертов — идентифицировать и оценить риски путем интервью подходящих квалифицированных специалистов. Специалисты высказывают своё мнение о рисках и дают им оценку, исходя из своих знаний, опыта и имеющейся информации. Этот метод может помочь избежать повторного наступления на одни и те же грабли.  
Перед опросом эксперт должен получить всю необходимую вводную информацию. Деятельность экспертов необходимо направлять, задавая вопросы. Во время опроса вся информация, выдаваемая экспертом, должна записываться и сохраняться. При работе с несколькими экспертами выходная информация обобщается и доводится до сведения всех задействованных экспертов.

***Мозговой штурм.***К участию в мозговом штурме привлекаются квалифицированные специалисты, которым дают «домашнее задание» — подготовить свои суждения по определенной категории рисков. Затем проводится общее собрание, на котором специалисты по очереди высказывают свои мнения о рисках. Важно: споры и замечания не допускаются. Все риски записываются, группируются по типам и характеристикам, каждому риску дается определение. Цель — составить первичный перечень возможных рисков для последующего отбора и анализа.

***Метод Дельфи.*** Метод Дельфи во многом похож на метод мозгового штурма. Однако есть важные отличия. Во-первых, при применении этого метода эксперты участвуют в опросе анонимно. Поэтому результат характеризуется меньшей субъективностью, меньшей предвзятостью и меньшим влиянием отдельных экспертов. Во-вторых, опрос экспертов проводится в несколько этапов. На каждом этапе модератор рассылает анкеты, собирает и обрабатывает ответы. Результаты опроса рассылаются экспертам снова для уточнения их мнений и оценок. Такой подход позволяет достичь некоего общего мнения специалистов о рисках.

***Карточки Кроуфорда.***Суть этой методики в следующем [[5]](http://citforum.ru/SE/project/arkhipenkov_lectures/10.shtml#ref.5.5). Собирается группа экспертов 7-10 человек. Каждому участнику мини-исследования раздается по десять карточек (для этого вполне подойдет обычная бумага для записок). Ведущий задает вопрос: "Какой риск является наиболее важным в этом проекте?" Все респонденты должны записать наиболее, по их мнению, важный риск в данном проекте. При этом никакого обмена мнениями не должно быть. Ведущий делает небольшую паузу, после чего вопрос повторяется. Участник не может повторять в ответе один и тот же риск.

После того как вопрос прозвучит десять раз, в распоряжении ведущего появятся от 70 до 100 карточек с ответами. Если группа подобрана хорошо (в том смысле, что в нее входят люди с различными точками зрения), вероятность того, что участники эксперимента укажут большинство значимых для проекта рисков, весьма высока. Остается составить список названных рисков и раздать его участникам для внесения изменений и дополнений.

### **4. Контрольные списки рисков проектов разработки ПО**

В качестве источника информации при выявлении рисков могут служить различные доступные контрольные списки рисков проектов разработки ПО, которые следует проанализировать на применимость к данному конкретному проекту. Например, Барии Боэм  приводит список 10 наиболее распространенных рисков программного проекта:

* Дефицит специалистов.
* Нереалистичные сроки и бюджет.
* Реализация несоответствующей функциональности.
* Разработка неправильного пользовательского интерфейса.
* "Золотая сервировка", перфекционизм, ненужная оптимизация и оттачивание деталей.
* Непрекращающийся поток изменений.
* Нехватка информации о внешних компонентах, определяющих окружение системы или вовлеченных в интеграцию.
* Недостатки в работах, выполняемых внешними (по отношению к проекту) ресурсами.
* Недостаточная производительность получаемой системы.
* "Разрыв" в квалификации специалистов разных областей знаний.

Демарко и Листер приводят свой список из пяти наиболее важных источников рисков любого проекта разработки ПО:

* Изъяны календарного планирования
* Текучесть кадров
* Раздувание требований
* Нарушение спецификаций
* Низкая производительность

Не существует исчерпывающих контрольных списков рисков программного проекта, поэтому необходимо внимательно анализировать особенности каждого конкретного проекта.

Результатом идентификации рисков должен стать список рисков с описанием их основных характеристик: причины, условия, последствий и ущерба.

***Пример.*** Если вернуться к примеру проекта создания «Автоматизированной системы продажи документации», который мы рассматривали в предыдущих лекциях, то список главных выявленных рисков может выглядеть следующим образом - таблица 3.

Таблица 3 Список рисков проекта создания «Автоматизированной системы продажи документации»

|  |  |  |  |
| --- | --- | --- | --- |
| **Причина** | **Условия** | **Последствия** | **Ущерб** |
| Требования не ясны. | Отсутствие описания сценариев использования системы. | Задержка начала разработки прикладного ПО. Большой объем переработок. | Задержки в сроках сдачи готового продукта и дополнительные трудозатраты. |
| Недостаток квалифицированных кадров. | Архитектура и код низкого качества. | Большое число ошибок. Большие затраты на их исправление. | Задержки в сроках сдачи готового продукта и дополнительные трудозатраты. |
| Текучесть кадров. | Частая смена участников команды. | Низкая производительность при вводе новых участников в проект. | Задержки в сроках сдачи готового продукта и дополнительные трудозатраты. |

За процессом идентификации рисков следует процесс их качественного анализа.

### 

### **5. Качественный анализ рисков**

***Качественный анализ рисков включает в себя расстановку рангов для идентифицированных рисков.***При анализе вероятности и влияния предполагается, что никаких мер по предупреждению рисков не производится.

***Качественный анализ рисков*** включает:

* Определение вероятности реализации рисков.
* Определение тяжести последствий реализации рисков.
* Определения ранга риска по матрице «вероятность — последствия».
* Определение близости наступления риска.
* Оценка качества использованной информации.

Для качественной оценки вероятности реализации риска и определения тяжести последствий его реализации применяется, как правило, общепринятые в организации шкалы, примеры которых мы приводили ранее.

Для определения ранга риска используется матрица вероятностей и последствий (Рисунок 10.3). **Ранг риска определяется произведением веса вероятности и значимости последствий** - трехуровневое ранжирование.



Рисунок 10.3. Ранг риска и матрица вероятностей и последствий

***Пример.*** Продолжая рассмотрение примера проекта создания «Автоматизированной системы продажи документации», матрица рангов главных выявленных рисков может выглядеть следующим образом (Таблица 4).

Таблица 4. Матрица рангов главных выявленных рисков проекта создания «Автоматизированной системы продажи документации»

|  |  |  |  |
| --- | --- | --- | --- |
| **Причина** | **Вероятность** | **Воздействие** | **Ранг** |
| Требования не ясны | Очень вероятно | Катастрофические | 9 |
| Недостаток квалифицированных кадров. | Очень вероятно | Критичные | 6 |
| Текучесть кадров. | Возможно | Критичные | 4 |

 Для оценки рисков необходима точная и адекватная информация. Использование неточной информации ведет к ошибкам в оценке. Неверная оценка риска также является риском.

**Критерии оценки качества используемой при анализе информации** выглядят следующим образом:

* Степень понимания риска.
* Доступность и полнота информации о риске.
* Надежность, целостность и достоверность источников данных.

Результатом качественного анализа рисков является их подробное описание (Таблица 5).

Таблица 5. Пример карточки с описанием риска

|  |  |
| --- | --- |
| **Номер:** R-101 | **Категория:** Технологический. |
| **Причина:** Недостаток квалифицированных кадров. | **Симптомы:** Разработчики будут использовать новую платформу — J2EE. |
| **Последствия:** Низкая производительность разработки | **Воздействие:** Увеличение сроков и трудоемкости разработки. |
| **Вероятность:** Очень вероятно. | **Степень воздействия:** Критичная. |
| **Близость:** Очень скоро. | **Ранг:** 6. |
| **Исходные данные:** «Содержание проекта», «План обеспечения ресурсами», Протоколы совещаний №21 от 01.06.20\_\_, №27 от 25.06.20\_\_. | |

Результаты качественного анализа используются в ходе последующего количественного анализа рисков и планирования реагирования на риски.

### **6. Количественный анализ рисков**

Количественный анализ производится в отношении тех рисков, которые в процессе качественного анализа были квалифицированы как имеющие высокий и средний ранг.

Для количественного анализа рисков могут быть использованы следующие методы:

* Анализ чувствительности.
* Анализ дерева решений.
* Моделирование и имитация.

**1) Анализ чувствительности** помогает определить, какие риски обладают наибольшим потенциальным влиянием на проект. В процессе анализа устанавливается, в какой степени неопределенность каждого элемента проекта отражается на исследуемой цели проекта, если остальные неопределенные элементы принимают базовые значения. Результаты представляются, как правило, в виде диаграммы «торнадо». Рисунок 10.4 представляет пример такой диаграммы, которая отражает влияние на проектные трудозатраты различных факторов профессионализма разработчиков ПО.

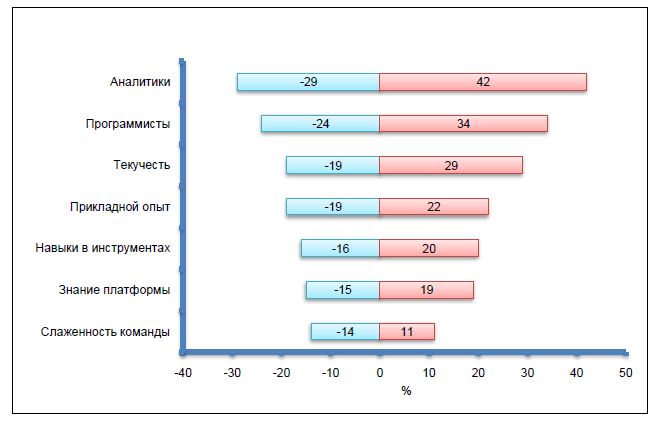


Рисунок 10.4. Влияние факторов профессионализма разработчиков ПО на трудозатраты по проекту.

2) Анализ дерева решений.

Анализ последствий возможных решений проводится на основе изучения диаграммы дерева решений, которая описывает рассматриваемую ситуацию с учетом каждой из имеющихся возможностей выбора и возможного сценария.

Рисунок 10.5 представляет пример диаграммы дерева решений на дугах которой проставлены вероятности и затраты при развитии событий по тому или иному сценарию. Критерием для принятия решения служит математическое ожидание потерь от его принятия.

### 

Рисунок 10.5. Пример анализ дерева решений при выборе покупать или производить необходимую для проекта библиотеку визуальных компонентов (VCL).

### 

### **7. Планирование реагирования на риски**

***Планирование реагирования на риски***— это процесс разработки путей и определения действий по увеличению возможностей и снижению угроз для целей проекта. Данный процесс начинается после проведения качественного и количественного анализа рисков.

Запланированные операции по реагированию на риски должны соответствовать серьезности риска, быть экономически эффективными в решении проблемы, своевременными, реалистичными в контексте проекта и согласованными со всеми участниками.

Возможны **четыре метода реагирования на риски**:

* Уклонение от риска (risk avoidance).
* Передача риска (risk transference).
* Снижение рисков (risk mitigation).
* Принятие риска (risk acceptance).

***1) Уклонение от риска***предполагает изменение плана управления проектом таким образом, чтобы исключить угрозу, вызванную негативным риском, оградить цели проекта от последствий риска или ослабить цели, находящиеся под угрозой (например, уменьшить содержание проекта). Некоторые риски, возникающие на ранних стадиях проекта, можно избежать при помощи уточнения требований, получения дополнительной информации или проведения экспертизы. Например, уклониться от риска можно, если отказаться от реализации рискованного функционального требования или самостоятельно разработать необходимый программный компонент, вместо ожидания поставок продукта от субподрядчика.

***2) Передача риска*** подразумевает переложение негативных последствий угрозы с ответственностью за реагирование на риск на третью сторону. Передача риска просто переносит ответственность за его управление другой стороне, но риск при этом никуда не девается. Передача риска практически всегда предполагает выплату премии за риск стороне, принимающей на себя риск. Например, заказ на стороне разработки рискованного компонента по фиксированной цене. В IT часто приходится формулировать риски в виде допущений, тем самым передавая его заказчику. Например, оценивая проект внедрения, мы можем записать допущение о том, что производитель не изменит стоимость лицензий на базовое ПО.

***3) Снижение рисков***предполагает понижение вероятности и/или последствий негативного рискованного события до приемлемых пределов. Принятие предупредительных мер по снижению вероятности наступления риска или его последствий часто оказываются более эффективными, нежели усилия по устранению негативных последствий, предпринимаемые после наступления события риска. Например, раннее разрешение архитектурных рисков снижает потери при досрочном закрытии проекта. Или регулярная ревизия поставок заказчиком может снизить вероятность риска его неудовлетворенности конечным результатом. Если в проектной команде высока вероятность увольнения сотрудников, то введение на начальной стадии в проект дополнительных (избыточных) людских ресурсов снижает потери при увольнении членов команды, поскольку не будет затрат на «въезд» в проектный контекст новых участников.

***4) Принятие риска*** означает, что команда проекта осознанно приняла решение не изменять план управления проектом в связи с риском или не нашла подходящей стратегии реагирования. Мы вынуждены принимать все «неизвестные риски». Принятие того, что всегда происходит, когда мы вообще не управляем рисками. Если же мы управляем рисками, то мы можем страховать риски, закладывая резерв в оценки срока завершения и/или трудозатрат. Проактивное отношение к принятым рискам может состоять в разработке план реагирования на риски. Этот план может быть введен в действие только при заранее определенных условиях, если есть уверенность и достаточное количество признаков того, что данный план будет успешно выполнен.

Важно помнить о вторичных рисках (Secondary Risks), возникающих в результате применения реагирования на риски, которые тоже должны быть идентифицированы, проанализированы и при необходимости включены в список управляемых рисков.

### **8. Главные риски программных проектов и способы реагирования**

***Главные причины провала программных проектов:***

* Требования заказчика отсутствуют / не полны / подвержены частым изменениям.
* Отсутствие необходимых ресурсов и опыта.
* Отсутствие рабочего взаимодействия с заказчиком.
* Неполнота планирования. «Забытые работы».
* Ошибки в оценках трудоемкостей и сроков работ.

К часто упускаемым требованиям можно отнести:

* Функциональные
* Программы установки, настройки, конфигурации.
* Миграция данных.
* Интерфейсы с внешними системами.
* Справочная система.
* Общесистемные
* Производительность.
* Надежность.
* Открытость.
* Масштабируемость.
* Безопасность.
* Кросплатформенность.
* Эргономичность.

Как правило, эти требования «всплывают» при подготовке и проведении приемо-сдаточных испытаний и могут сильно задержать проект по времени и увеличить трудозатраты на его реализацию. Чтобы этого не происходило, следует достигать соглашения с заказчиком по всем перечисленным пунктам предпочтительнее еще на стадии инициации проекта.   
Если вероятность изменений требований проекта высока, то возможны следующие подходы для реагирования на данный риск:

* Переоценка проекта каждый раз, когда требования добавляются / изменяются (уклонение).
* Итерационная разработка. Контракт с компенсацией затрат на основе «Time & Materials» (передача риска Заказчику).
* Учет в оценках трудоемкости и сроков возможности роста требований, например, на 50% (резервирование риска).

И еще, при сборе требований следует соблюдать принцип минимализма Вольтера: ***«Рассказ закончен не тогда, когда в него нечего добавить, а тогда, когда из него нечего больше выкинуть»***. Для большинства программных продуктов применим принцип Парето: 80% ценности продукта заключены лишь в 20% требований к нему.

Если у нас в проекте недостаточно квалифицированных специалистов, то мы можем снизить последствия этого риска, применив следующие действия:

* Привлечь экспертов-консультантов на начальных этапах.
* Учитывать в оценках трудоемкости издержки на обучение сотрудников.
* Уменьшать потери от текучести кадров, привлекая на начальном этапе избыточное число участников.
* Учесть в оценках «время разгона» для новых сотрудников.

Для установления открытых и доверительных отношений с заказчиком, необходимо предпринимать следующие шаги:

* Постоянное взаимодействие.
* Согласование пользовательских интерфейсов и разработка прототипа продукта.
* Периодические поставки тестовых версий конечным пользователям для их оценки.

При планировании работ по проекту часто «забывают»:

* Обучение.
* Координация работ.
* Уточнение требований.
* Управление конфигурациями.
* Разработка и поддержка скриптов автосборки.
* Разработка автотестов.
* Создание тестовых данных.
* Обработка запросов на изменения.

И еще. Не стоит надеяться, что участники проекта будут каждую неделю по 40 часов работать именно над вашим проектом. Есть множество причин, по которым они не смогут работать по проекту 100% своего времени. К списку наиболее распространенных причин этого относятся:

* Сопровождение действующих систем.
* Повышение квалификации.
* Участие в подготовке технико-коммерческих предложений.
* Участие в презентациях.
* Административная работа.
* Отпуска, праздники, больничные.

Рекомендация, планировать, что разработчики, которые назначены в ваш проект на 100% будут реально работать над вашими задачами в среднем от 24 до 32 часов в неделю.

### **9. Управление проектом, направленное на снижение рисков**

На стадии инициации проекта оценка его трудоемкости имеет погрешность от -50% до +100% . Это, если оценка хорошая! А если плохая, то неопределенность, а, следовательно, и риски сорвать сроки и превысить плановую трудоемкость, могут быть в разы больше. Если не прилагать специальных усилий этот «дамоклов меч» неопределенности будет висеть над проектом на всем его протяжении (Рисунок 10.6).

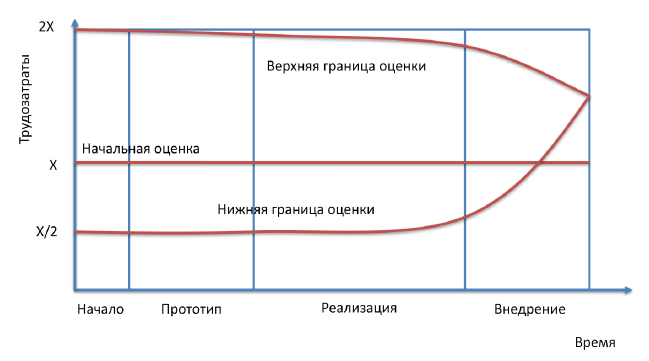


Рисунок 10.6. Неопределенность не уменьшается, если управление не направлено на раннее разрешение рисков

***Проектом следует управлять так, чтобы риски несвоевременной сдачи и перерасхода ресурсов постоянно снижались.***

Ранее мы уже говорили о том, что 80% ценности разработки обусловлена лишь 20% требований к продукту, без реализации которых продукт для заказчика становится просто ненужным. Остальные требования, как правило, так называемые «украшательства», от части которых заказчик, как правило, может отказаться, чтобы получить проект в срок. Поэтому **следует в первую очередь реализовывать ключевые функциональные требования**.

Но есть и еще **архитектурные риски**. Известно, что закон Парето применим и к потреблению вычислительных ресурсов: 80% потребления ресурсов (время и память) приходится на 20% компонентов. Поэтому, необходимо реализовывать архитектурно-значимые требования так же в первую очередь, создавая «представительный» прототип будущей системы, который «простреливает» весь стек, применяемых технологий. Прототип позволит измерить и оценить общесистемные свойства будущего продукта: доступность, быстродействие, надежность, масштабируемость и проч. (Рисунок 10.7)  
Ошибка — реализовывать сначала легкие требования, чтобы продемонстрировать быстрый прогресс проекта.



Рисунок 10.7. Определение приоритетов требований на первые итерации проекта

Управление, нацеленное на снижение рисков, позволяет существенно снизить неопределенность на ранних стадиях проекта (Рисунок 10.8).

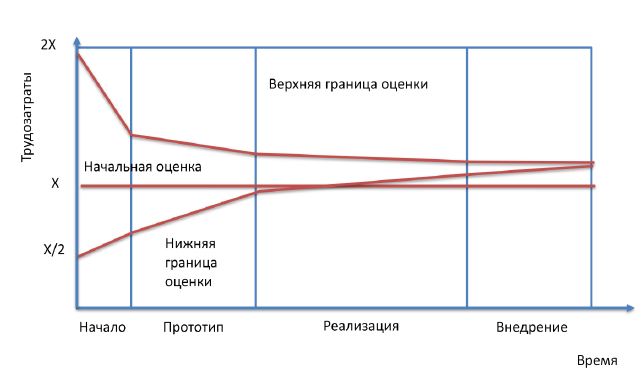


Рисунок 10.8. Управление, нацеленное на снижение рисков, позволяет уменьшать неопределенность

**Проработка ключевых функциональных требований и детальное планирование их реализации** позволяет уменьшить разброс начальных оценок, примерно, в 2 раза: от -30% до +50%.

**Детальное проектирование и разработка прототипа будущей системы** позволит получить еще более точные оценки общей трудоемкости: от -10% до +15%.

Может оказаться так, что по результатам прототипирования, уточненные оценки суммарной трудоемкости окажутся неприемлемыми. В этом случае проект придется закрыть досрочно, но потери при этом, будут значительно меньше, чем в случае, если то же самое произойдет, когда проект уже в 2 раза превысит первоначальную оценку трудоемкости.

Если с заказчиком не удается найти взаимоприемлемое решение при первоначальной оценке проекта, то разумно попытаться договориться о выполнении проекта в 2 этапа с самостоятельным финансированием:

* Исследование. Бизнес-анализ, уточнение требований, проектирование и прототипироание решения, уточнение суммарных оценок трудозатрат. Эта работа, как правило, требует 10% общих трудозатрат и 20% времени всего проекта.
* Непосредственно реализация. Если уточненные оценки трудозатрат окажутся приемлемыми для заказчика.

С вменяемыми заказчиками это часто удается.

### **10. Мониторинг и контроль рисков**

Управление рисками должно осуществляться на протяжении всего проекта. Не вести мониторинг рисков в ходе проекта — все равно, что не следить за уровнем топлива при поездке на автомобиле.

***Мониторинг и управление рисками***— это процесс идентификации, анализа и планирования реагирования на новые риски, отслеживания ранее идентифицированных рисков, а также проверки и исполнения операций реагирования на риски и оценка эффективности этих операций.  
В процессе мониторинга и управления рисками используются различные методики, например, анализ трендов и отклонений, для выполнения которых необходимы количественные данные об исполнении, собранные в процессе выполнения проекта.

***Мониторинг и управления рисками включает в себя следующие задачи***:

* Пересмотр рисков.
* Аудит рисков.
* Анализ отклонений и трендов.

***Пересмотр рисков***должен проводиться регулярно, согласно расписанию. Управление рисками проекта должно быть одним из пунктов повестки дня всех совещаний команды проекта. Неплохо начинать каждый статус митинг с вопроса: «Ну и какие еще неприятности нас ожидают?» Идентификация новых рисков, и пересмотр известных рисков происходит с использованием процессов, описанных ранее.

***Аудит рисков***предполагает изучение и предоставление в документальном виде результатов оценки эффективности мероприятий по реагированию на риски, относящихся к идентифицированным рискам, изучение основных причин их возникновения, а также оценку эффективности процесса управления рисками.

***Тренды в процессе выполнения проекта*** подлежат проверке с использованием данных о выполнении. *Для мониторинга выполнения всего проекта могут использоваться анализ освоенного объема и другие методы анализа отклонений проекта и трендов.* На основании выходов этих анализов можно прогнозировать потенциальные отклонения проекта на момент его завершения по показателям стоимости и расписания. Отклонения от базового плана могут указывать на последствия, вызванные как угрозами, так и благоприятными возможностями.

### **Выводы**

Отказываться от управления проектными рисками это все равно, что в кинотеатре не иметь огнетушителей и плана эвакуации на случай пожара.

Все, что мы делаем, управляя проектом разработки ПО, должно быть направлено на борьбу с рисками не уложиться в срок, перерасходовать ресурсы, разработать не тот продукт, который требуется.

Цели управления рисками проекта — снижение вероятности возникновения и/или значимости воздействия неблагоприятных для проекта событий.

Главные причины провала программных проектов:

* Требования заказчика отсутствуют / не полны / подвержены частым изменениям.
* Отсутствие необходимых ресурсов и опыта.
* Отсутствие рабочего взаимодействия с заказчиком.
* Неполнота планирования. «Забытые работы».
* Ошибки в оценках трудоемкостей и сроков работ.

***Основная литература***

1. Архипенков С. Лекции по управлению программными проектами// Электронный ресурс:http://citforum.ru/SE/project/

2. Основы управления ИТ-проектами [Электронный ресурс]: учебное пособие / сост. Е.Р. Кирколуп, Ю.Г., Скурыдин, Е.М. Скурыдина. – Барнаул : АлтГПУ, 2017// Электронный ресурс:http://library.altspu.ru/dc/pdf/kirkolup.pdf

3. Рыбалова Е.А. Управление проектами: учебное пособие / Е.А. Рыбалова. - Томск: ФДО, ТУСУР, 2015. - 205 с.

4. Богданов В.В. Управление проектами в Microsoft Project 2007. Учебный курс. -  СПб: Питер, 2010

[5. Коптелов А., Никитин И., Цулая М. Гибкая процессная методология Agile](https://www.intuit.ru/studies/courses/3590/832/info). - М.: ИНТУИТ // Электронный ресурс: // www.intuit.ru/studies/courses/3590/832/print\_lecture/31170

6. Топ-7 методов управления проектами: Agile, Scrum, Kanban, PRINCE2 и другие// www.pmservices.ru/project-management-news/top-7-metodov-upravleniya-proektami-agile-scrum-kanban-prince2-i-drugie/

### 7. [Грекул](https://www.intuit.ru/intuituser/userpage/v_i_grekul) В., [Коровкина](https://www.intuit.ru/intuituser/userpage/n_l_korovkina) Н., [Куприянов](https://www.intuit.ru/intuituser/userpage/y_v_kupriyanov) Ю. [Методические основы управления ИТ-проектами](https://www.intuit.ru/studies/courses/646/502/info). - М.: ИНТУИТ // Электронный ресурс: https://www.intuit.ru/studies/courses/646/502/info

### ***Дополнительная литература:***

8. Сериков А. Основы управления проектами. - М.: ИНТУИТ. Электронный ресурс: www.intuit.ru/studies/professional\_retraining/965/courses/272/info

9. Управление проектами: Учебное пособие / М.В. Романова. - М.: ИД ФОРУМ: НИЦ ИНФРА-М, 2014

10. Руководство к Своду знаний по управлению проектами РМВОК (A Guide to the Project Management Body of Knowledge PMBOK®) — 4-е изд. — Project Management Institute, Inc., 2008. — 464 с.

11. https://products.office.com/

12. Том ДеМарко, Тимоти Листер, «Вальсируя с Медведями. Управление рисками в проектах по разработке программного обеспечения», М., Компания p.m.Office, 2005

**Лекция 11. Управление качеством**

*План:*

1. Качество проекта

2. Планирование качества

3. Обеспечение качества

4. Контроль качества

**1. Качество проекта**

*Управление качеством* проекта включает в себя процессы и действия исполняющей организации, политику в области качества, цели и сферы ответственности в области качества таким образом, чтобы проект удовлетворял тем нуждам, ради которых он был предпринят. Осуществляется посредством системы управления качеством, предусматривающей определенные правила и процедуры, а также действия по постоянному совершенствованию процессов, проводимые на всем протяжении проекта. **Процессы управления качеством проекта, включают в себя**:

1. *Планирование качества* — процесс определения требований и/или стандартов качества для проекта и продукта и документирования соответствия установленным требованиям и стандартам.

2. *Обеспечение качества* — процесс проверки соблюдения требований к качеству и результатов измерений в процессе контроля качества.

3. *Контроль качества* — процесс контроля и записи результатов выполнения действий по обеспечению качества. Оценка исполнения и разработки рекомендаций относительно необходимых изменений.

Эти процессы взаимосвязаны друг с другом и с процессами других областей знаний. **Управление качеством проекта направлено как на управление проектом, так и на продукт проекта**. Невыполнение требований к качеству продукта или проекта может привести к серьезным отрицательным последствиям для заинтересованных сторон проекта.

*Модель управления качеством* в основе своей соответствует требованиям Международной организации по стандартизации (International Organization for Standardization, **ISO**).

Эта модель учитывает также авторские модели управления качеством, разработанные Демингом (Deming), Юраном (Juran), Кросби (Crosby) и др., и общие модели, такие как тотальное управление качеством (**TQM**), Шесть сигм (Six Sigma), анализ характера и последствий отказов, контрольные оценки на этапе проектирования, мнение заказчика, стоимость качества (COQ) и постоянное совершенствование.

Современное управление качеством служит дополнением к управлению проектом. Важны следующие *положения*:

• *Удовлетворенность потребностей заказчика*— понимание, оценка, определение и управление ожиданиями заказчика, чтобы его требования оказались выполненными. Обеспечить *сочетание* соответствия требованиям и пригодности к использованию.

• *Предотвращение важнее проверок* — качество должно обеспечиваться за счет планирования, разработки и производства, а не за счет проведения инспекций. Затраты на предупреждающие действия по предотвращению ошибок, значительно ниже, чем стоимость их исправления после обнаружения в результате проверок.

• *Постоянное совершенствование* — цикл «планирование-выполнение-проверка-действие» (модель, описанная Шухартом и усовершенствованная Демингом) является основой для повышения качества. Среди моделей совершенствования процессов (рис. 11.1) можно привести Организационную модель зрелости управления проектами (Organizational Project Management Maturity Model, OPM3®) и интегрированную модель развития функциональных возможностей (Capability Maturity Model Integrated, CMMI®).

• *Ответственность руководства* — за обеспечение проекта ресурсами, необходимыми для его успешного завершения, ответственность несет руководство. *Стоимость качества* обозначает общую стоимость всех мероприятий, направленных на обеспечение качества, на протяжении жизненного цикла продукта.

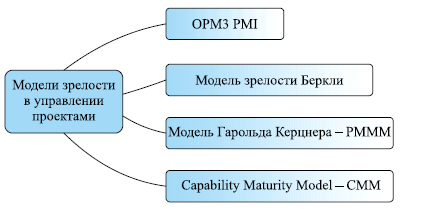


Рисунок 11.1. Модели зрелости организации УП

**2. Планирование качества**

***Планирование качества*** — *процесс* ***определения требований*** *и/или стандартов качества для проекта и продукта, а также документирования того, каким образом проект будет демонстрировать соответствие установленным требованиям и стандартам.*

***Планирование качества: инструменты и методы***

1. *Сравнительный анализ затрат и выгод*.

Основная выгода от выполнения требований к качеству может заключаться в уменьшении числа доработок, увеличении производительности, уменьшении затрат и росте удовлетворенности заинтересованных сторон проекта. В экономическом обосновании каждого действия в области качества сравнивается стоимость соответствующей меры в отношении качества с ожидаемой от нее выгодой.

2. *Стоимость качества* — это совокупная стоимость всех мероприятий на протяжении жизненного цикла продукта, направленных на повышение качества, обеспечение соответствия определенным требованиям, а также предупреждение факторов, способных вызвать снижение качества и его несоответствие требованиям (доработка). Издержки вследствие дефектов разделяют на внутренние (выявленные в рамках проекта) и внешние (выявленные заказчиком).

3. *Контрольные карты* используются для определения стабильности процесса, характеризуется ли он предсказуемым выполнением. Нижние и верхние границы, заданные спецификацией, основаны на *требованиях контракта*. Они отражают максимальные и минимальные допустимые значения. Могут налагаться штрафы, связанные с превышением границ, заданных спецификацией. Процесс считается вышедшим из-под контроля в том случае, если точка данных находится за контрольными границами или если семь последовательных точек находятся выше или ниже средней линии.

4. *Бенчмаркинг* предусматривает *сопоставление* текущего или планируемого проекта с другими сопоставимыми проектами с целью выявления лучших практик, выработки идей для совершенствования и определения критериев оценки исполнения. Другие сопоставимые проекты могут быть как внутри исполняющей организации, так и за ее пределами, а также могут относиться к аналогичной прикладной области или иной.

5. *Планирование экспериментов (ПЭ)* — это статистический метод, позволяющий определить факторы, способные оказывать влияние на конкретные параметры продукта или процесса в ходе разработки или производства. Одним из важных аспектов данного метода является статистическая система, предназначенная для анализа систематических изменений всех важных факторов. Анализ экспериментальных данных должен способствовать разработке оптимальных условий для продукта или процесса, обнаружению факторов, оказывающих влияние на результаты, и выявлению взаимодействий между факторами.

6. *Выборочные оценки* предусматривают выбор части *интересующей совокупности* для проверки (например, произвольный выбор десяти инженерных чертежей из восьмидесяти). Частота и размеры выборок должны определяться в ходе процесса планирования качества, чтобы в стоимости качества учитывался ряд испытаний, ожидаемые отходы и т. д.

7. *Дополнительные инструменты планирования качества* используются для лучшего *определения требований* к качеству и планирования действий по эффективному управлению качеством. Включают методы:

• *Мозговой штурм*.

• *Диаграммы сходства*, используемые для визуального определения логических группировок, основанных на естественных взаимосвязях.

• *Анализ силовых полей*, представляющий собой диаграммы сил, выступающих за и против изменения.

• *Матричные диаграммы* содержат две, три или четыре группы информации и показывают взаимосвязи между факторами, причинами и целями.

Данные в матрице организованы в строки и столбцы с пересекающимися ячейками, которые могут быть заполнены информацией, описывающей взаимосвязь между элементами, расположенными в строке и столбце.

• *Матрицы расстановки приоритетов*, позволяющие ранжировать набор разнообразных проблем и/или вопросов (обычно генерируемых с помощью мозгового штурма) по уровню их важности.

**3. Обеспечение качества**

***Обеспечение качества*** *представляет собой* ***процесс проверки соблюдения требований*** *к качеству и результатов измерений в процессе контроля качества для обеспечения использования соответствующих стандартов и метрик качества.*

*Обеспечение качества* — один из процессов исполнения, в котором используются данные, полученные во время контроля качества, составляет основу для постоянного совершенствования процессов, итеративных мер по улучшению качества всех процессов, сокращает трату ресурсов.

*Аудит качества* — это структурированная, *независимая проверка*, определяющая, насколько операции проекта соответствуют установленным в рамках проекта или организации правилам, процессам и процедурам. *Целями* аудита качества являются: выявление хороших/лучших применяемых практик и распространение внедренных; выявление всех узких мест/недостатков; внесение достижений аудита в хранилище накопленных знаний.

***Управление временем, содержанием и стоимостью -*** Согласование мер по обеспечению выполнения плана управления качеством, учета стоимости и достаточного количества ресурсов для их проведения. Этот процесс является критически важным и в случае его невыполнения произойдет неоправданное увеличение стоимости проекта и сроков его выполнения. Владельцем процесса является руководитель проекта.

**4. Контроль качества**

***Контроль качества*** *представляет собой процесс* ***контроля и записи*** *результатов действий, направленных на обеспечение качества, для оценки исполнения и разработки* ***рекомендаций*** *относительно необходимых изменений.*

Контроль качества осуществляется на протяжении всего проекта, выявляет причины низкого качества процессов или продуктов. К результатам проекта относятся *результаты работ*, *управленческие результаты*, показатели выполнения *стоимости и сроков*.

***Контроль качества: инструменты и методы***

Первые семь из данных инструментов и методов известны как «*семь основных инструментов качества Ишикавы*».

1. *Причинно-следственные диаграммы* — также называемые диаграммами Ишикавы или диаграммами «*рыбий скелет*» — иллюстрируют связь различных факторов с возможными проблемами и следствиями (рис. 11.2).

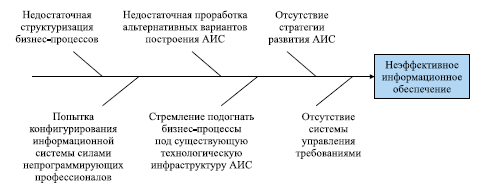


Рисунок 11.2. Диаграмма Ишикавы, причины и следствия

Возможную первопричину можно выявить, постоянно задавая вопросы «*почему*?» или «*как*?» по мере движения вдоль одной из линий. При анализе первопричин могут быть использованы диаграммы «почему-почему» и «как-как». Причинно-следственные диаграммы также используются при анализе рисков.

2. *Контрольные карты*, предназначены для сбора и анализа соответствующих данных с целью определения *статуса качества* процессов и продуктов проекта.

Контрольные карты дают наглядное представление о развитии процесса во времени, представляют собой графическое отображение отклонения процесса и дают ответ на вопрос: «*Находится ли отклонение данного процесса в рамках установленных границ*?». Вносить коррективы в процесс следует тогда, когда процесс выходит за рамки установленных границ. Семь последовательных точек за пределами верхней или нижней контрольной границы указывают на процесс, который вышел из-под контроля. Верхняя и нижняя контрольные границы обычно устанавливаются в пределах ±3 сигмы, где 1 сигма — одно среднее квадратичное отклонение.

3. *Гистограммы* — это вертикальные столбиковые диаграммы, отображающие распределение переменных (параметр или свойство проблемы). Высота каждого столбика обозначает относительную частоту свойства. Данный инструмент иллюстрирует наиболее частую причину возникновения проблем процесса по количеству и относительной высоте столбиков. Рисунок 11.3 является примером *неупорядоченной* гистограммы, показывающей причины *позднего ввода времени* командой проекта.



Рисунок 11.3. Столбиковая диаграмма причины проблем

4. *Диаграмма Парето* является особым типом гистограммы, *упорядоченной по частоте* возникновения, и показывает, какое количество обнаруженных дефектов является следствием причин, относящихся к определенному типу или категории (рис. 11.4).

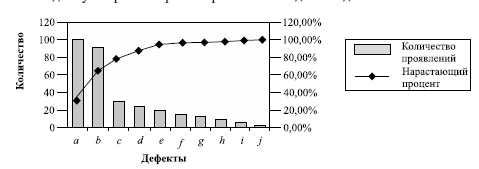


Рисунок 11.4. Диаграмма Парето

Диаграмма Парето концептуально связана с *законом Парето*, который утверждает, что относительно *малое число причин обычно вызывает большинство проблем* или дефектов. Этот закон также известен как **принцип 80/20**, согласно которому *80% проблем вызвано 20% причин*. Диаграммы Парето также могут использоваться для суммирования разнообразных типов данных для анализа 80/20.

5. *Диаграмма тренда* отражает историю и характер изменений, представляет собой линейный график, отображающий точки данных, расположенных на графике в порядке их возникновения, дает представление о тенденциях, колебаниях во времени, а также о позитивных и негативных изменениях процесса во времени.

*Анализ тенденций* проводится с помощью диаграмм тренда и включает в себя использование *математических методов для прогнозирования* будущих результатов на основе данных прошлых периодов. Диаграмма тренда часто используется для наблюдения за показателями: *техническое исполнение* (сколько ошибок, дефектов выявлено, не исправлено?); *выполнение стоимости и сроков* (сколько операций выполнено со значительными отклонениями в каждый период времени?).

6. *Диаграмма разброса* показывает взаимосвязь между двумя переменными (рисунок 11.5).

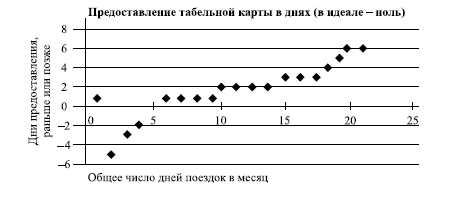


Рисунок 11.5. Диаграмма разброса

Данный инструмент позволяет команде контроля качества изучить и определить возможную взаимосвязь между изменениями, наблюдаемыми в обеих переменных. На графике против зависимых переменных изображаются независимые переменные. Чем ближе друг к другу точки на диагональной линии, тем более тесно они связаны.

7. *Выборочные оценки.* Порядок отбора и проверки выборок определен в плане качества.

8. *Инспекция* — это проверка продукта работы для определения его соответствия документированным *стандартам*. Как правило, результаты инспекции содержат результаты измерений. Например, *инспекция* (проверка, экспертная оценка, аудит или сквозной контроль) результатов может проводиться по отдельной операции или по конечному продукту проекта.

9. *Проверка одобренных запросов на изменение* — для подтверждения того, что они внедрены именно так, как было одобрено.

Мониторинг результатов проекта для установления соответствию стандартам качества. Определение и устранение причин, вызывающих отклонения. Последствием невыполнения данного процесса будет отклонение от ожидаемых результатов, причину которого невозможно установить и исправить. Контроль качества является важным процессом и отвечает за него команда проекта или руководитель проекта.

Обеспечение качества – процесс выполнения плановых систематических операций по качеству, которые обеспечивают выполнение всех предусмотренных процессов, необходимых для того, чтобы проект соответствовал установленным требованиям по качеству. Функцию обеспечения качества может выполнять команда проекта, руководство исполняющей организации, заказчик или спонсор, другие участники проекта. Для контроля качества проекта проводятся аудиторские проверки, целью которых является выяснение соответствия качества проекта стандартам, установленным в плане обеспечения качества. Процесс обеспечения качества включает методы непрерывного улучшения качества будущих проектов.

Аудиты качества проводятся на основе критериев, каждый из которых является следствием требований нормативной документации системы менеджмента качества (требование ISO 9000) и системы управления проектами (PMBOK).

***Основная литература***

1. Архипенков С. Лекции по управлению программными проектами// Электронный ресурс:http://citforum.ru/SE/project/

2. Основы управления ИТ-проектами [Электронный ресурс]: учебное пособие / сост. Е.Р. Кирколуп, Ю.Г., Скурыдин, Е.М. Скурыдина. – Барнаул : АлтГПУ, 2017// Электронный ресурс:http://library.altspu.ru/dc/pdf/kirkolup.pdf

3. Рыбалова Е.А. Управление проектами: учебное пособие / Е.А. Рыбалова. - Томск: ФДО, ТУСУР, 2015. - 205 с.

4. Богданов В.В. Управление проектами в Microsoft Project 2007. Учебный курс. -  СПб: Питер, 2010

[5. Коптелов А., Никитин И., Цулая М. Гибкая процессная методология Agile](https://www.intuit.ru/studies/courses/3590/832/info). - М.: ИНТУИТ // Электронный ресурс: // www.intuit.ru/studies/courses/3590/832/print\_lecture/31170

6. Топ-7 методов управления проектами: Agile, Scrum, Kanban, PRINCE2 и другие// www.pmservices.ru/project-management-news/top-7-metodov-upravleniya-proektami-agile-scrum-kanban-prince2-i-drugie/

### 7. [Грекул](https://www.intuit.ru/intuituser/userpage/v_i_grekul) В., [Коровкина](https://www.intuit.ru/intuituser/userpage/n_l_korovkina) Н., [Куприянов](https://www.intuit.ru/intuituser/userpage/y_v_kupriyanov) Ю. [Методические основы управления ИТ-проектами](https://www.intuit.ru/studies/courses/646/502/info). - М.: ИНТУИТ // Электронный ресурс: https://www.intuit.ru/studies/courses/646/502/info

### ***Дополнительная литература:***

8. Сериков А. Основы управления проектами. - М.: ИНТУИТ. Электронный ресурс: www.intuit.ru/studies/professional\_retraining/965/courses/272/info

9. Управление проектами: Учебное пособие / М.В. Романова. - М.: ИД ФОРУМ: НИЦ ИНФРА-М, 2014

10. Руководство к Своду знаний по управлению проектами РМВОК (A Guide to the Project Management Body of Knowledge PMBOK®) — 4-е изд. — Project Management Institute, Inc., 2008. — 464 с.

11. https://products.office.com/

**Лекция 12. Управление взаимодействием. Реализация проекта**

*План:*

1. Управление взаимодействием проекта

### 2. Реализация проекта

### 2.1 Рабочее планирование

### 2.2. Принципы количественного управления

### 2.3. Завершение проекта

**1. Управление взаимодействием проекта**

*Управление взаимодействием проекта (Project Communication Management)* — раздел управления проектами, включающий процессы, предназначенные для организации своевременного и приемлемого составления, сбора и распределения, хранения и конечного использования проектной информации. Состоит из процессов планирования взаимодействий, распределения информации, отчетности по исполнению и административного завершения.

*Управление коммуникациями*  проекта включает в себя процессы, необходимые для своевременного создания, сбора, распространения, хранения, получения и использования информации проекта.

Эффективные коммуникации являются мостом, связывающим различные заинтересованные стороны, вовлеченные в проект, объединяя разнообразные культурные и организационные особенности, различные уровни опыта, а также различные взгляды и интересы в отношении выполнения или результата проекта. Общая схема *управления коммуникациями* проекта включает в себя *процессы*:

*1. Определение заинтересованных сторон проекта* — процесс выявления всех людей или организаций, на которых будет оказывать влияние проект, документирования значимой информации относительно их интересов, вовлеченности и влияния на успех проекта.

2. *Планирование коммуникаций* — процесс выявления потребностей заинтересованных сторон проекта в информации и определения подхода к коммуникациям.

3. *Распространение информации* — процесс предоставления информации заинтересованным сторонам проекта в соответствии с планом.

4. *Управление ожиданиями заинтересованных сторон проекта* — процесс общения и работы с заинтересованными сторонами проекта с целью удовлетворения их потребностей и решения возникающих проблем.

5. *Отчеты об исполнении* — процесс сбора и распространения информации об исполнении, включая отчеты о текущем состоянии, оценку исполнения и прогнозы.

Для того чтобы проектные коммуникации наиболее эффективно решали задачи, стоящие перед проектом, еще на фазе планирования проекта необходимо четко сформулировать стратегию коммуникаций. Наиболее важным элементом планирования коммуникаций является идентификация получателей информации, далее следует озаботиться планированием содержания информационных сообщений, которое может значительно меняться в зависимости от адресата. Затем происходит выбор канала коммуникации и определение отправителя. За фактическим выполнением коммуникаций необходимо всегда осуществлять обратную связь, позволяющую корректировать наши действия в дальнейшем.

Относительно содержания любое сообщение на проекте должно включать в себя информацию следующего рода.

1. *Удовлетворение потребности участников проекта понимать.*Участники проекта должны иметь возможность получить объективную, полную и непротиворечивую информацию о целях и задачах проекта и иметь возможность сформировать собственное рациональное мнение о проекте.

2. *Удовлетворение потребности участников проекта чувствовать.* Заинтересованные стороны должны четко понимать, какие процедуры предусмотрены для организации их участия в принятии решений по проекту, есть ли каналы обратной связи, как они могут быть вовлечены в реализацию наиболее значимой для них части проекта.

3. *Удовлетворение потребности участников проекта действовать.* Сотрудники должны быть проинформированы, какие средства, методы, инструменты предусмотрены для их скорейшей адаптации в условиях новой организационно-функциональной среды бизнеса организации.

Рассмотрим пример стратегии коммуникации.

**Стратегия коммуникации должна содержать:**

1. Цели и задачи информирования участников проекта.

Например, в рамках реализуемой стратегии информирование не сводится исключительно к обеспечению сотрудников необходимой информацией о проекте. Цели информирования направлены на повышение лояльности персонала компании к проекту, что служит достижению конечной цели управления изменениями – обеспечению безболезненного перехода к новому бизнес-стандарту.

Перед процессом информирования стоят три основных цели:

● обеспечение целевой аудитории информацией о целях, задачах и результатах проекта:

○ проинформировать сотрудников компании о проекте, его важности и выгодах;

○ обеспечить доступность, корректность и своевременность получения информации о проекте;

○ поддерживать интерес к проекту на всех фазах его реализации;

● обеспечение целевой аудитории информацией об осведомленности проектной команды о предстоящих изменениях в функциональной среде и организационной структуре компании, и о проблемах, связанных с ними:

○ обеспечить своевременность получения информации о предстоящих изменениях целевыми аудиториями;

○ сформировать образ проекта как открытой, прозрачной и доступной инициативы;\_\_

○ реализовать сбор сведений об ожиданиях/опасениях бизнес-экспертов о предстоящих изменениях;

● обеспечение целевой аудитории информацией о планах по переходу к новым процессам, обязанностям, методам работы:

○ сформировать образ проекта как инициативы, готовой поддерживать обратную связь, отвечать на вопросы и помогать в решении предстоящих проблем;

○ обеспечить заблаговременное информирование конечных пользователей о предстоящих мероприятиях, связанных с управлением изменениями;

2. Роли и обязанности. Указание конкретных лиц, ответственных за проектные коммуникации, и их места в организационной структуре проекта.

3. Целевую аудиторию.

**2. Реализация проекта**

### **2.1 Рабочее планирование**

***Управление — это расчленение, анализ, определение последовательности действий, конкретная реализация.***

Управление фокусируется на нижнем уровне: как мне сделать это наилучшим образом? Эта компетенция руководителя определяет эффективность движения по выбранному пути. Как правило, менеджеры, вышедшие из программистов, без особого труда овладевают необходимыми управленческими навыками.

***Базовое расписание, составленное на этапе планирования проекта, служит ориентиром для мониторинга состояния дел на макроуровне.***

***Для оперативного управления проектом используется рабочий план.***

***Рабочее планирование рекомендуется выполнять методом «набегающей волны»:*** **работа, которую надо будет выполнить в ближайшей перспективе, подробно планируется на низшем уровне ИСР, а далеко отстоящая работа планируется на сравнительно высоком уровне ИСР.**

Элементарная работа, как правило, представляет собой отдельное функциональное требование к программному продукту или запрос на изменение, над которым последовательно работают: бизнес-аналитик, проектировщик, разработчик, тестировщик и документалист.

***Трудоемкость элементарной работы каждого из исполнителей должна быть от 4 до 20 чел.\*час.***Если трудоемкость задачи не укладывается в эти пределы, следует провести декомпозицию работы.

***Для рабочего планирования целесообразно использовать систему управления задачами или багтрекинга,*** поскольку она позволяет задавать последовательность переходов задачи от исполнителя к исполнителю, управлять приоритетами работ и адекватно отслеживать их статус: анализ, проектирование, кодирование, тестирование, документирование.

***Работа должна считаться законченной только тогда, когда реализация требования протестирована и документирована.***

**В зависимости от уровня профессионализма и зрелости команды проекта распределение работ может осуществляться либо директивно с жесткой постановкой срока и контролем исполнения каждой задачи, либо эти полномочия делегируются исполнителям.**В этом случае они сами выбирают задачи последовательно в соответствие с приоритетами, а их выполнение анализируется периодически на статус митинге.

Можно рекомендовать еженедельные собрания по статусу проекта всей команды или, если проект достаточно большой, то ключевых его частников: руководителей подпроектов и лидеров команд. Хорошее время для этого утро понедельника, поскольку участники проекта, особенно студенты, которые совмещают учебу и работу, часто работают в выходные, но, разумеется, не потому, что аврал, а потому что им так удобнее. Обсуждаются, как правило, всего три вопроса:

* Угрозы и проблемы.
* Анализ результатов за неделю.
* Уточнение приоритетов задач на новую неделю.

Как правило, н***ет смысла оценивать процент реализации работы в промежуточном состоянии***, поскольку, если задача передана в тестирование, то это вовсе не означает, что 70% работы сделаны. На этапе тестирования может быть выявлена ошибка проектирования и вся работа начнется заново.

***Рекомендация — использовать правило «50/100»***. Если работа по задаче начата, то следует учитывать ее, как выполненную на 50%. А 100% получает только протестированная и документированная работа.

### **2.2. Принципы количественного управления**

«Тем, что нельзя измерить, нельзя управлять». Измерения по проекту необходимо выполнять регулярно, не реже одного раза в 1-2 недели. Для каждого измеримого показателя должны быть определены его плановые значения.

***Для каждого планового значения должны быть определены три области критичности отклонений:***

* Допустимые отклонения. Предполагается, что никаких управляющих воздействий не требуется.
* Критичные отклонения. Требуется тщательный анализ причин отклонения и при необходимости применение корректирующих действий.
* Недопустимые отклонения. Требуется срочный анализ причин отклонения и обязательное применение корректирующих действий.

***Измерения необходимо производить регулярно.***Цель — выявить причины наступивших или возможных критичных и недопустимых отклонений. Результатом анализа должны стать планирование корректирующих действий по компенсации недопустимых отклонений, их реализация и мониторинг результативности применения этих корректирующих действий.

***Все измерения необходимо сохранять в репозитарии проекта.***Измерения, накопленные в ходе проекте, являются наиболее достоверной основой при детальной оценке и планировании работ на следующих итерациях проекта.

Поскольку ***главная задача менеджера удержать проект в пределах «железного» треугольника, то, в первую очередь, необходимо анализировать отклонения проекта по срокам и затратам.*** Делается это ***при помощи метода освоенного объема***[[1]](http://citforum.ru/SE/project/arkhipenkov_lectures/15.shtml#ref.8.1). Приходилось сталкиваться с мнением, что этот метод не применим в управлении программными проектами. Это действительно так, если мы используем «водопадную» модель процесса разработки. Но если ИСР проекта, ориентирована на инкрементальную разработку, то это означает, что на верхних уровнях декомпозиции находятся компоненты проектного продукта и их функционал, а не производственные процессы. Следовательно, если в проекте реализованы, протестированы и документированы 50 % функциональных требований, то есть все основания полагать, что осталась приблизительно половина проектных работ.

***Суть метода оценки проекта по освоенному объему*** заключается в следующем.

Сначала оценивается***отклонение от графика*SV**(Shedule Variance) в денежных единицах:

**SV = EV - PV,**

где

* **EV** (Earned Value) — освоенный объем. Плановая стоимость выполненных работ. Объем выполненных работ, выраженный в терминах одобренного бюджета, выделенного на эти работы для плановой операции и элемента иерархической структуры работ;
* **PV** (Planned Value) — плановый объем. Плановая стоимость запланированных работ. Утвержденный бюджет, выделенный на плановые работы, выполняемые в рамках плановой операции или элемента иерархической структуры работ.

Например. Пусть мы на текущий момент реализовали (протестировали и документировали) 20 функциональных требований, на каждое из которых было запланировано затратить по 40 чел.\*час. по 1000 руб, то освоенный объем будет

**EV = 20 \* 40 \* 1000 = 800 000 руб.**

Если же на текущий момент планировалось реализовать только 15 требований, то плановый объем будет

**PV = 15 \* 40 \* 1000 = 600 000 руб.**

Следовательно, мы опережаем график (отклонение от графика положительное) на величину

**SV = EV - PV = 800 000- 600 000 = 200 000 руб.**

Как пересчитываете отклонение от графика, выраженное в денежных единицах, в сокращение сроков проекта иллюстрируется на Рисунке 12.1.

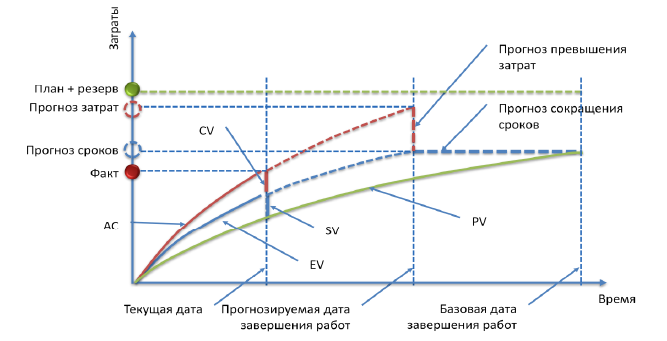


Рисунок 12.1. Оценка и прогноз показателей по методу освоенного объема

Если мы опережаем график, то это не обязательно означает что проект идет успешно. Хорошо это или плохо зависит от значения другого показателя метода освоенного объема: **CV** (Cost Variance) — отклонения по затратам, которое оценивается по формуле:

**CV = EV - AC**

где

* **AC** (Actual Cost) — фактические затраты. Фактическая стоимость выполненных работ. Фактические затраты на выполнение работ за определенный период в рамках плановой операции или элемента иерархической структуры работ.

Например, если мы для того что сократить время работ по проекту работали 25% времени сверхурочно и в выходные дни с двойной оплатой, то фактические трудозатраты составили:

**AC = 20 \* (30 \* 1000 + 10 \* 2000) = 1 000 000 руб.**

Поэтому отклонения по затратам в нашем случае будет  
**CV = EV - AC PV = 800 000 - 1 000 000 = - 200 000 руб.**

Отрицательное значение отклонения по затратам означает, что мы превысили бюджет, что, в общем случае, не очень хорошо. Но если срок завершения проекта для нас имеет высший приоритет, и наши прогнозируемые затраты по завершению проекта не превышают плановых с учетом управленческого резерва (Рисунок 12.1), то в этом случае можно считать, что проект выполняется успешно.

Отклонение от бюджета и по срокам в абсолютных денежных единицах недостаточно для характеристики проектов разных масштабов.

Более наглядны относительные показатели: **индекс выполнения сроков SPI**(Schedule Performance Index)

**SPI = EV / PV**

и **индекс выполнения стоимости CPI**(Cost Performance Index)

**CPI = EV/ AC,**

которые характеризуют проект независимо от его размера. Если значения обоих индексов больше 1, то это свидетельствуют о благополучном состоянии в проекте.

Какие еще ***измеримые показатели целесообразно применять в управлении программным проектом***?

В первую очередь это **показатель прогресса проекта**, доля реализованных и проверенных высокоуровневых требований к проекту, например, отношение числа завершенных сценариев использования продукта к их общему числу.

Другой показатель — **стабильность проекта**, общее количество принятых (утвержденных спонсором или заказчиком) изменений в плане управления проектом. Чем выше нестабильность в проекте, тем больше сложность в управлении работами и ниже производительность участников.

Всегда следует измерять текущий **размер проекта** — количество строк исходного кода, добавленных, измененных и удаленных в ходе выполнения проекта разработки ПО. Чем больше объем исходного кода, тем больше времени потребуется на внесение изменений и исправление ошибок.  
При увеличении объема проектного продукта трудозатраты на каждую новую строку исходного кода увеличиваются. Если за номинал взять производительность проектной команды при производстве продукта в 10 KSLOC, то та же команда на проекте в 100 KSLOC покажет производительность в 1.3–1.7 раз меньшую, а на проекте в 1000 KSLOC следует ожидать, что производительность снизится в 1.6–3.0 раза.

Большой объем кода так же потребует большего количества людей на его сопровождение. Поскольку, даже если будет выявляться только несколько критических ошибок в год, то для того чтобы их исправить в приемлемые сроки, например, за 24 часа, в продукте общим объемом в 1000 KSLOC то один программист с этим не справится. Это связано с тем, что для того чтобы исправить ошибку в ограниченные сроки необходимо оперативно выявить и устранить ее причину, а для этого надо хорошо знать архитектуру и код программного продукта. Чтобы эффективно сопровождать продукт подобного объема необходимо иметь в «горячем» резерве примерно 20 разработчиков, потому что 50 KSLOC, на мой взгляд, это предельный объем кода, который может удерживать в голове и эффективно сопровождать один человек. И еще проблема: чем этих людей занимать в свободное от исправлений ошибок время, если нет новых проектов развития продукта.

Следующий важный показатель состояния проекта — это **средняя производительность**, отношение текущего размера проекта к фактическим затратам по проекту. С. Макконнелл  приводит следующие показатели (минимальное, максимальное и среднее значение) производительности в KSLOC на один чел.\*мес. фактических затрат для стандартных типов проектов объемом в 100 KSLOC:

* 300-7000 (800) — интранет система.
* 200-7000 (600) — бизнес система.
* 100-2000 (300) — Интернет система.
* 50-600 (100) — системное ПО, телекоммуникации.
* 20-300 (50) — системы реального времени.

Высокая производительность в проекте — это далеко не всегда хороший признак. Приходилось встречаться с проектами, в которых вследствие активного применения метода «copy+past», средняя производительность в разработке бизнес системы достигала 2000 SLOC/чел.\*мес. Однако для реализации требуемого функционала было написано в 3–4 раза больше кода, чем это могло бы потребоваться при адекватной проработке архитектуры.

Еще одна группа количественных показателей, которые следует наблюдать в ходе реализации проекта, характеризует **качество программного продукта**:

* ***Дефектность продукта***— количество выявленных дефектов на единицу объема продукта (например, KSLOC).
* ***Доля неустраненных дефектов*** — отношение количества незакрытых максимально критичных и критичных дефектов к количеству выявленных несоответствий.
* ***Средние затраты на сопровождение*** — средние трудозатраты на исправление одного дефекта. Высокое значение этого показателя может свидетельствовать о некачественной архитектуре программного продукта.
* ***Документированность кода*** — определяет процент строк исходного кода с комментарии по отношению к общему количеству строк.

Следует подчеркнуть, что наблюдать надо за средними по проекту значениями показателей, и ни в коем случае не пытаться измерять индивидуальные характеристики производительности и качества. Главные причины, почему это не следует делать, заключаются в том, что, во-первых, в этом случае вместо слаженной командной работы мы получим личную конкуренцию, а, во-вторых, наиболее «продвинутые» разработчики станут работать на формальные показатели, а не на достижение целей проекта.  
Если команда действительно состоялась, то для нее характерна коллективная ответственность за достижение общих целей. И, как пишет, Т.Демарко [[3]](http://citforum.ru/SE/project/arkhipenkov_lectures/15.shtml#ref.8.3), «менеджер проекта должен занимать очередь, чтобы покритиковать сотрудника, не выполняющего свои обещания», поскольку в правильной команде для этого всегда найдется масса желающих.

### **2.3. Завершение проекта**

***Главная цель этой фазы — проверить и передать заказчику результат проекта.*** Для этого необходимо выполнить приемо-сдаточные работы в соответствии с процедурой приемки, которая должна быть определена заранее на самой ранней стадии проекта.

Результаты проекта должны быть переданы во внедрение или сопровождение, или должным образом законсервированы для дальнейшего использования. Не должно оставаться «зависших» работ по проекту. Все линейные руководители всех участников должны быть извещены о завершении работ по проекту, и освобождении сотрудников.

***Важная задача, которая должна быть решена на данной фазе, это реализация обратной связи по проекту.***Цель — сохранить результаты, знания и опыт, полученные в проекте, для более эффективного и качественного выполнения аналогичных проектов в будущем. Необходимо архивировать все результаты, документировать опыт, уроки по проекту и предложения по улучшению технологии выполнения работ и управления проектами.  
Все проекты и в особенности провальные проекты должны завершаться итоговым отчетом, если компания не хочет «наступать на одни и те же грабли». Помним о том, что «вчерашние проблемы, это сегодняшние риски».

***Итоговый отчет должен содержать следующую информацию:***

* **Итоги проекта:**
* **Достижение целей проекта**
* **Дополнительные полезные результаты**
* **Фактические сроки**
* **Фактические расходы**
* **Обоснование отклонения от целей**
* **Отклонения результатов от требований**
* **Уроки проекта**
* **Проблемы проекта и способы их решения**
* **Материалы программные компоненты для последующего использования**
* **Предложения по изменению технологий или стандартов компании**

На фазе завершения желательно реализовать и план мотивации участников проектной команды, поскольку отложенное вознаграждение мотивирует существенно слабее.

### **Выводы**

Для оперативного управления проектом используется рабочий план.

Элементарная работа, как правило, представляет собой отдельное функциональное требование к программному продукту или запрос на изменение, над которым последовательно работают: бизнес-аналитик, проектировщик, разработчик, тестировщик и документалист.

Измерения по проекту необходимо выполнять регулярно, не реже одного раза в 1–2 недели. Для каждого измеримого показателя должны быть определены его плановые значения и допустимые отклонения.

В состав измеряемых показателей должны входить следующие характеристики проекта:

* Освоенный и плановый объемы работ и фактические затраты по проекту.
* Показатели прогресса и стабильности проекта.
* Размер продукта.
* Производительность.
* Показатели качества программного продукта.

По результатам проекта обязательно должна быть реализована обратная связь. Цель — сохранить результаты, знания и опыт, полученные в проекте, для более эффективного и качественного выполнения аналогичных проектов в будущем.

***Основная литература***

1. Архипенков С. Лекции по управлению программными проектами// Электронный ресурс:http://citforum.ru/SE/project/

2. Основы управления ИТ-проектами [Электронный ресурс]: учебное пособие / сост. Е.Р. Кирколуп, Ю.Г., Скурыдин, Е.М. Скурыдина. – Барнаул : АлтГПУ, 2017// Электронный ресурс:http://library.altspu.ru/dc/pdf/kirkolup.pdf

3. Рыбалова Е.А. Управление проектами: учебное пособие / Е.А. Рыбалова. - Томск: ФДО, ТУСУР, 2015. - 205 с.

4. Богданов В.В. Управление проектами в Microsoft Project 2007. Учебный курс. -  СПб: Питер, 2010

[5. Коптелов А., Никитин И., Цулая М. Гибкая процессная методология Agile](https://www.intuit.ru/studies/courses/3590/832/info). - М.: ИНТУИТ // Электронный ресурс: // www.intuit.ru/studies/courses/3590/832/print\_lecture/31170

6. Топ-7 методов управления проектами: Agile, Scrum, Kanban, PRINCE2 и другие// www.pmservices.ru/project-management-news/top-7-metodov-upravleniya-proektami-agile-scrum-kanban-prince2-i-drugie/

### 7. [Грекул](https://www.intuit.ru/intuituser/userpage/v_i_grekul) В., [Коровкина](https://www.intuit.ru/intuituser/userpage/n_l_korovkina) Н., [Куприянов](https://www.intuit.ru/intuituser/userpage/y_v_kupriyanov) Ю. [Методические основы управления ИТ-проектами](https://www.intuit.ru/studies/courses/646/502/info). - М.: ИНТУИТ // Электронный ресурс: https://www.intuit.ru/studies/courses/646/502/info

### ***Дополнительная литература:***

8. Сериков А. Основы управления проектами. - М.: ИНТУИТ. Электронный ресурс: www.intuit.ru/studies/professional\_retraining/965/courses/272/info

9. Управление проектами: Учебное пособие / М.В. Романова. - М.: ИД ФОРУМ: НИЦ ИНФРА-М, 2014

10. Руководство к Своду знаний по управлению проектами РМВОК (A Guide to the Project Management Body of Knowledge PMBOK®) — 4-е изд. — Project Management Institute, Inc., 2008. — 464 с.

11. https://products.office.com/

12. Том ДеМарко, Тимоти Листер, «Вальсируя с Медведями. Управление рисками в проектах по разработке программного обеспечения», М., Компания p.m.Office, 2005

13. Макконнелл С. Сколько стоит программный проект. - Спб: Питер», 2007.

14. Демарко Т., Листер Т. Человеческий фактор: успешные проекты и команды. - Спб: Символ-Плюс, 2005.