

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ
ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ОРЛОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ И.С. ТУРГЕНЕВА»

А.Ю. Ужаринский, О.В. Конюхова, П.В. Лукьянов

**УПРАВЛЕНИЕ ПРОГРАММНЫМИ ПРОЕКТАМИ В IT КОМПАНИЯХ.
ЭТАПЫ ИНИЦИАЛИЗАЦИИ И ПЛАНИРОВАНИЯ**

Орёл
ОГУ имени И.С. Тургенева
2020

УДК 004.2(075.8)
ББК 32.971.32-02я7
К 655

Рецензенты:

доктор технических наук, профессор кафедры информационных систем
федерального государственного бюджетного
образовательного учреждения высшего образования
«Орловский государственный университет имени И.С. Тургенева»
А.В. Коськин,

кандидат экономических наук, доцент
зав. кафедрой «Информационные технологии» ФГБОУ ВО «БГИТУ»
О.Д. Казаков

Ужаринский А.Ю.,

Управление программными проектами в IT компаниях. Этапы инициализации и планирования / А.Ю. Ужаринский, О.В. Конюхова, П.В. Лукьянов – Орёл: ОГУ имени И.С. Тургенева, 2020. – 78 с.

Учебное пособие содержит актуальную информацию о методах управления проектами разработки программного обеспечения в IT компаниях: история развития методов управления проектами, основные процессы управления, организационные структуры компании, инструменты, применяемые для управления различными компонентами проекта, а также современное состояние отрасли управления в организационных системах.

Рассматриваются основные понятия теории управления, структура проекта, этапы жизненного цикла проекта, методы оценки экономической эффективности и сроков реализации проекта, основные документы, оформляемые на начальных этапах управления проектами в компании.

Учебное пособие соответствует требованиям ФГОС и содержанию программ учебных дисциплин «Основы управления программными проектами» для студентов направлений 09.03.01 «Информатика и вычислительная техника», 09.03.02 «Информационные системы и технологии», 09.03.03 «Прикладная информатика», 09.03.04 «Программная инженерия».

Предназначено для студентов направлений 09.03.01 «Информатика и вычислительная техника», 09.03.02 «Информационные системы и технологии», 09.03.03 «Прикладная информатика», 09.03.04 «Программная инженерия». Может быть полезно студентам других специальностей при изучении методов управления проектами в различных сферах профессиональной деятельности.

УДК 004.2(075.8)
ББК 32.971.32-02.7

© ОГУ имени И.С. Тургенева, 2020

Содержание

Введение.....	4
1 Управление проектами. Основные определения и концепция	5
1.1 Понятие проекта и управления проектами.....	5
1.2 Классификация и типы проектов.....	8
1.3 Структура проекта.....	11
1.4 Жизненный цикл проекта	17
1.5 Организационная структура компании.....	18
1.6 История развития методов управления проектами	25
Контрольные вопросы	28
2 Фаза инициации проекта	30
2.1 Работы, выполняемые на фазе инициации проекта	30
2.2 Модель видения проекта	31
2.3 Пример видения программного продукта	37
2.4 Структура концепции проекта.....	40
2.5 Концепция проекта разработки программного обеспечения ..	48
Контрольные вопросы.	52
3 Оценка трудоёмкости и экономической эффективности проекта	53
3.1 Оценка экономической эффективности проекта	53
3.2 Общие принципы оценки трудоёмкости проекта.....	59
3.3 Оценка трудоёмкости проекта по методике PERT.....	61
3.4 Метод функциональных точек для оценки трудоёмкости проекта.....	63
3.5 Оценка трудоёмкости проекта разработки программного обеспечения на основе методики COSOMO II	72
Контрольные вопросы.	77
Литература	78

Введение

В своей ежедневной деятельности человек постоянно сталкивается с необходимостью выполнения проектов различной сложности. Их эффективное выполнение невозможно без применения методов управления проектами. В современной экономике эффективное функционирование компании без применения методов управления проектами практически не возможно. Особенно актуально это утверждение для IT компаний, занимающихся разработкой программного обеспечения.

Учебное пособие содержит актуальную информацию о методах управления проектами разработки программного обеспечения в IT компаниях: история развития методов управления проектами, основные процессы управления, организационные структуры компании, инструменты, применяемые для управления различными компонентами проекта, а также современное состояние отрасли управления в организационных системах.

Рассматриваются основные понятия теории управления, структура проекта, этапы жизненного цикла проекта, методы оценки экономической эффективности и сроков реализации проекта, основные документы, оформляемые на начальных этапах управления проектами в компании.

Учебное пособие соответствует требованиям ФГОС и содержанию программ учебных дисциплин «Основы управления программными проектами» для студентов направлений 09.03.01 «Информатика и вычислительная техника», 09.03.02 «Информационные системы и технологии», 09.03.03 «Прикладная информатика», 09.03.04 «Программная инженерия».

1 Управление проектами. Основные определения и концепция

1.1 Понятие проекта и управления проектами

В классической теории управления выделяется два вида человеческой деятельности – операционная и проектная. Под операционной деятельностью понимается циклически повторяющаяся деятельность, сроки завершения которой не имеют явного окончания. Условия выполнения такой деятельности заранее известны и не изменяются, а работы чётко регламентированы и повторяются в каждой итерации. Примером такой деятельности может быть серийное производство продукции на предприятии в соответствии с установленным технологическим процессом. Производство каждой новой партии продукции предполагает выполнение аналогичных действий, которые прописаны в её технологическом процессе.

Если же деятельность предполагает решение новой задачи или создание нового продукта и её операции чётко не регламентированы, то такая деятельность называется проектной. Для такой деятельности характерно отсутствие заранее известной структуры работ. Каждый новый проект это уникальное предприятие, отличающееся от других проектов, выполняемых в компании. Деятельность по разработке программного обеспечения в IT компаниях носит в большей степени проектный характер, т.к. каждый программный продукт не похож на программные продукты, разрабатываемые в компании ранее и обладает уникальными признаками.

Термин проект применяется в различных сферах деятельности и отраслях знаний в связи с этим существует множество определений термина проекта, фокусирующих внимание на различных аспектах данного понятия. Рассмотрим несколько типовых определений понятия проекта:

Слово **«прое́кт»** происходит от латинского «projectus», что означает «брошенный вперёд, выступающий, выдающийся вперёд».

Ю. И. Попов и О. В. Яковенко дают следующее определение понятия **проект** – это некоторое предприятие с изначально

установленными целями, достижение которых определяет завершение проекта. [1]

А. А. Бовин, Л. Е. Чередникова, В. А. Якимович предлагают определять **проект** как деятельность, мероприятие, предполагающие осуществление комплекса каких-либо действий, обеспечивающих достижение определённых целей.[3]

Проект – некоторая задача с определенными исходными данными и требуемыми результатами (целями), обуславливающими способ ее решения.[2]

Согласно стандарту по управлению проектами ISO 21500 под **проектом** необходимо понимать уникальный набор процессов, состоящих из скоординированных и управляемых задач с начальной и конечной датами, предпринятых для достижения цели.[4]

Проект включает в себя замысел (проблему), средства его реализации (решения проблемы) и получаемые в процессе реализации результаты. (Рис.1)

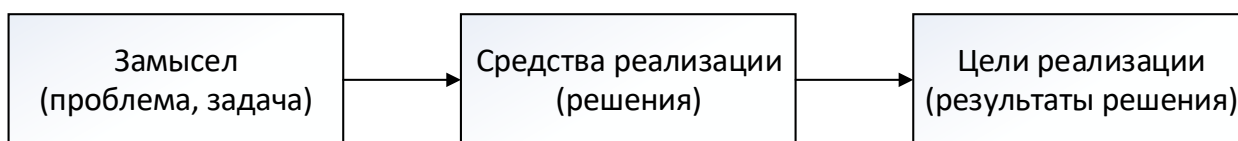


Рис. 1. Компоненты проекта

На основе приведённых определений понятия проекта можно выделить следующие основные признаки проекта:

1. направленность на достижение конкретных целей;
2. координированное выполнение взаимосвязанных операций;
3. сложность структуры проекта;
4. ограничения по времени и ресурсам;
5. уникальность (неповторимость).

С точки зрения системного подхода **проект** – ограниченное по времени организованное определённым образом целенаправленное изменение отдельной системы, ограниченное бюджетом всех видов ресурсов и содержащее конкретные требования к параметрам конечного результата. [5] В данном определении проект рассматривается как некоторая система взаимосвязанных элементов. Любая система характеризуется своей

структурой. Основными структурными элементами проекта являются:

1. Дерево целей и результатов;
2. Бюджет проекта;
3. Матрица распределения работ во времени и по исполнителям;
4. Сетевая модель проекта;
5. Матрица распределения и минимизации рисков;
6. График обеспечения ресурсами;
7. График финансирования проекта;
8. Матрица распределения ответственности;
9. Структурная декомпозиция контрактов;
10. Структурная модель организации проекта.

Деятельность, направленная на разработку, управление и выполнение проектов называется управлением проектами. Существуют следующие определения понятия управление проектами.

Управление проектом – искусство руководства и координации людских и материальных ресурсов на протяжении жизненного цикла проекта путем применения системы современных методов и техники управления для достижения определенных в проекте результатов по составу и объему работ, стоимости, времени, качеству и удовлетворению участников проекта. [2]

Управление проектами – это приложение знаний, опыта, методов и средств к работам проекта для удовлетворения требований, предъявляемых к проекту, и ожиданий участников проекта. Чтобы удовлетворить этим требованиям и ожиданиям, необходимо найти оптимальное сочетание между целями, сроками, затратами, качеством и другими характеристиками проекта. [6]

Методы управления проектами позволяют:

- 1 определить цели проекта и провести его обоснование, его жизнеспособность;
- 2 выявить структуру проекта (подцели, задачи, работы которые предстоит выполнить);
- 3 определить необходимые объемы и источники финансирования;
- 4 подобрать исполнителей – в частности, через процедуры торгов и конкурсов;

- 5 подготовить и заключить контракты;
- 6 определить сроки реализации проекта, составить график его выполнения, рассчитать необходимые ресурсы;
- 7 рассчитать смету и бюджет проекта;
- 8 планировать и учитывать риски;
- 9 обеспечить контроль за ходом реализации проекта и др.

1.2 Классификация и типы проектов

В связи с тем, что понятие проекта используется в различных сферах деятельности и описывает широкий круг процессов, существует множество различных вариантов классификации проектов. Рассмотрим основные типы проектов. В зависимости от направленности выделяют:

1. Технические
2. Организационные
3. Социальные
4. Экономические
5. Смешанные

Технические проекты связаны с разработкой нового продукта или технического изделия. Основными характерными чертами этих проектов являются:

1. Чёткая сформулированность цели проекта. При этом отдельные подцели проекта могут корректироваться по мере его реализации.
2. Сроки реализации проекта известны заранее на момент начала его выполнения. Сроки могут корректироваться в процессе выполнения проекта.
3. Расходы и бюджет проекта фиксированы и не изменяются в процессе его реализации.
4. Ограничения при реализации проекта связаны чаще всего с лимитом ресурсов, доступных для его реализации.

Организационные проекты связаны с реформированием структуры и процессов на предприятии или созданием новых предприятий и организаций, а также осуществлением некоторых мероприятий. Характерные черты таких проектов:

1. Цели проекта чётко определены, но зачастую не имеют точных количественных оценок. Это связано с тем, что такие

проекты направлены на улучшение неких организационных и системных процессов.

2. Продолжительность проектов и сроки их реализации задаются заранее и не могут сдвигаться.

3. В проект вовлечено большое число участников и заинтересованных сторон.

4. Расходы на проект подвергаются тщательному контролю и часто происходит корректировка плана расходов по ходу выполнения проекта.

Экономические проекты направлены на реструктуризацию предприятий, приватизацию, санацию, совершенствование таможенных правил, налоговой системы и др. Их характерные черты:

1. На момент начала проекта известны лишь приблизительные цели их реализации. В процессе выполнения проекта цели могут корректироваться.

2. Сроки выполнения проектов оцениваются приблизительно и часто изменяются в процессе реализации проекта.

3. Расходы определены приблизительно, но они жёстко контролируются во время реализации проекта.

Социальные проекты направлены на решение определённых социальных проблем. К таким проектам можно отнести проекты совершенствования пенсионного законодательства, ликвидации последствий социальных и природных катастроф и т.п. Характеристики таких проектов:

1. Цели проекта задаются в общих чертах и часто меняются по ходу реализации проекта. При этом сложно провести количественную оценку целей проекта.

2. Сроки реализации проекта не возможно определить заранее. Часто они представляют собой некоторую вероятностную оценку.

3. Ресурсы на проект выделяются по мере необходимости.

Смешанные проекты затрагивают несколько сфер деятельности.

По масштабу различают следующие классы проектов:

1. Малые.

2. Средние.

3. Крупные.

Малые проекты зачастую не велики по объёму (например, в США: капиталовложения до \$10-15 млн.; трудозатраты до 40-50 тыс. чел-час), имеют простую внутреннюю структуру и небольшое число участников. Ему присущи следующие характерные черты:

1. Координация работ осуществляется одним лицом;
2. Члены команды проекта являются взаимозаменяемыми;
3. Максимально простые графики реализации
4. Каждый член команды знает свой объём работ.

Крупные проекты – это целевые проекты, которые состоят из множества более мелких проектов, объединённых одной общей целью, временем осуществления и ресурсами. Это могут быть региональные, межрегиональные, национальные, отраслевые и межотраслевые проекты. Объёмы финансирования таких проектов составляют сотни миллионов долларов. Сроки реализации составляют от 5-7 лет и больше. Координируются такие проекты на государственном уровне. Характерными чертами таких проектов являются:

1. Наличие множества различных исполнителей специализирующихся на разных видах деятельности;
2. Обязательный учёт социальных и экономических условий реализации таких проектов.
3. Выделение в самостоятельную стадию разработку концепции проекта
4. Необходимость создания разных планов реализации проекта
5. Обязательный учёт и управление рисками проекта
6. Постоянный мониторинг хода реализации проекта.

По длительности реализации проекты делятся на:

1. Краткосрочные. Длительностью до 2 лет.
2. Среднесрочные. Длительностью от 2 до 5 лет.
3. Долгосрочные. Длительностью свыше 5 лет.

По сложности проекты делятся на:

1. Простой
2. Сложный
3. Уникальный

По характеру участников проекты делятся на:

1. Отечественные
 - 1.1 государственные
 - 1.2 региональные

1.3 местные

2. Зарубежные

3. Совместные

По количеству привлечённых других проектов:

1. Монопроекты

2. Мультипроекты

3. Мегапроекты

Монопроект представляет собой отдельный проект различного масштаба, вида и типа

Мультипроект – комплексный, он включает в свой состав несколько монопроектов.

Мегапроект представляет собой целевую программу развития, включающую в себя несколько моно и мультипроектов.

В качестве примера рассмотрим проект разработки мобильного приложения, реализующего афишу кинотеатра. По направленности данный проект техническим. По масштабу проект является малым. По длительности – краткосрочный. По сложности – простой. По характеру участников – местный. По количеству привлечённых других проектов – монопроект.

Существует множество других вариантов классификации проектов в зависимости от выбранного основания и целей классификации.

1.3 Структура проекта

Структура проекта – это организация связей и отношений между ее элементами. Программные проекты, как правило, имеют иерархическую переменную структуру, которая формируется применительно к условиям функционирования.

Структура проекта призвана определить продукцию, которую необходимо разработать или произвести, и связывает элементы работы, которые предстоит выполнить, – как между собой, так и с конечной целью проекта.

Применительно к реальным проектам, структура проекта должна сочетать разделение на:

1. компоненты структуры проекта;
2. этапы жизненного цикла;
3. элементы организационной структуры.

Процесс выделения структуры проекта направлен на разбиение проекта на отдельные поддающиеся управлению компоненты, определение связей между этими компонентами и составление плана выполнения проекта. Основными задачами структуризации проекта являются следующие задачи:

1. Разбиение проекта на поддающиеся управлению блоки.
2. Составление плана выполнения проекта.
3. Закрепление ответственных за выполнение работ и увязка работ с организационной структурой компании.
4. Оценка затрат, необходимых для реализации проекта.
5. Увязка работ проекта с системой бухгалтерских счетов, применяемых в организации.
6. Переход от общих целей проекта к конкретным задачам и работам, которые должны быть решены в проекте для достижения поставленных целей.

Рассмотрим подробнее составляющие структуры проекта. Процесс определения структуры проекта включает в себя ряд последовательных этапов (рис. 2) в рамках которых формируются отдельные компоненты проекта.



Рис. 2. Этапы планирования и реализации проекта

1. Определение целей проекта. Процесс структуризации начинается с формулирования цели проекта и определения всех

конечных продуктов проекта. Цель должна быть чёткой, однозначной, достижимой и измеримой. Кроме формулировки основной цели проект может содержать ряд сопутствующих подцелей, которые должны быть достигнуты в процессе выполнения проекта. В результате выстраивается иерархия целей проекта. Основным инструментом, применяемыми на данном этапе структуризации проекта является дерево целей проекта (рис. 3) Дерево целей проекта – это граф, показывающий как основная цель проекта разбивается на подцели на более глубоких уровнях детализации.



Рис. 3. Дерево целей проекта

2. Определение уровня детализации проекта. На этом этапе рассматриваются различные уровни детализации планирования и описания проекта, и определяется оптимальное количество уровней детализации структуры проекта.

3. Установка структуры процесса. На этом этапе выбирается модель жизненного цикла процесса, и определяются этапы реализации проекта.

4. Изучение организационной структуры проекта. На этом этапе изучаются все группы лиц, заинтересованных и участвующих в реализации проекта. К ним относятся как окружение проекта, так и участники команды проекта. Основным инструментом, применяемым для описания организационной структуры проекта, является дерево организационной структуры проекта – Organisation Breakdown Structure (ORG) (рис. 4). ORG – представляет собой ролевую модель команды проекта, описанную в виде иерархии подотчётности. Каждая роль участника проекта определяется набором полномочий и ответственности.

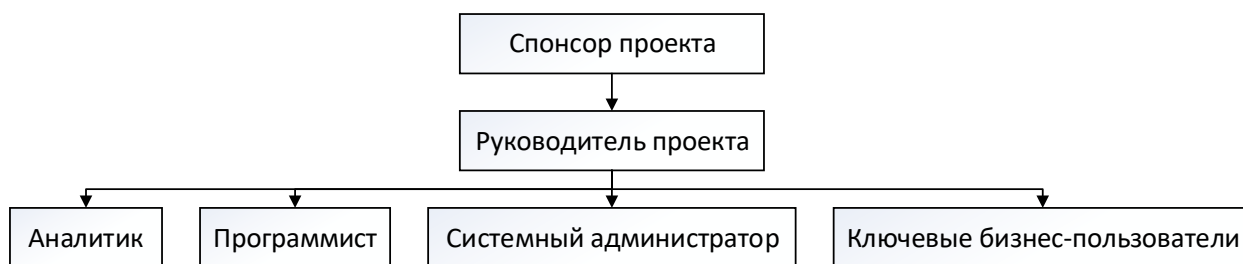


Рис. 4. Дерево организационной структуры проекта

5. Разработка структуры продукции. Структура продукции предполагает разбивку продукта по подсистемам и компонентам, включая материальное обеспечение, программное обеспечение, информационное обеспечение и сервисное обеспечение. Основным инструментом, применяемым на данном этапе является дерево структуры продукции Product Breakdown Structure (PBS) (рис. 5). Дерево структуры продукции – это граф, показывающий из каких компонентов будет состоять итоговая продукция проекта.

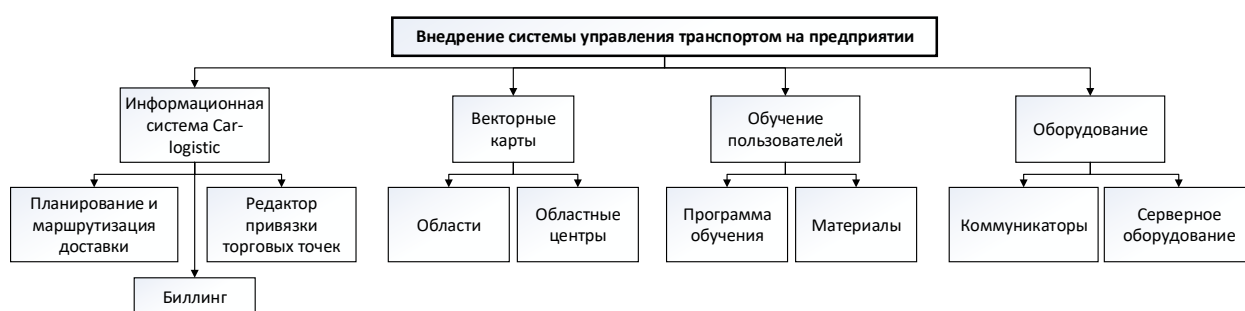


Рис. 5. Дерево структуры продукции.

6. Анализ системы бухгалтерских счетов в компании. Данный этап предполагает разработку систему правил ведения бухгалтерского учёта всех операций, выполняемых в рамках проекта. Эти правила должны основываться на существующий в организации системе бухгалтерского учёта.

7. Объединение схемы разбивки проекта в единую модель. Разработанные на этапах 3,4,5 модели объединяются на данном этапе в единую модель. Основным инструментом, применяемым на данном этапе является иерархическая структура работ Work Breakdown Structure (WBS) (рис. 6). Иерархическая структура работ это дерево операций, которые необходимо выполнить для реализации компонентов продукции и достижения целей проекта.

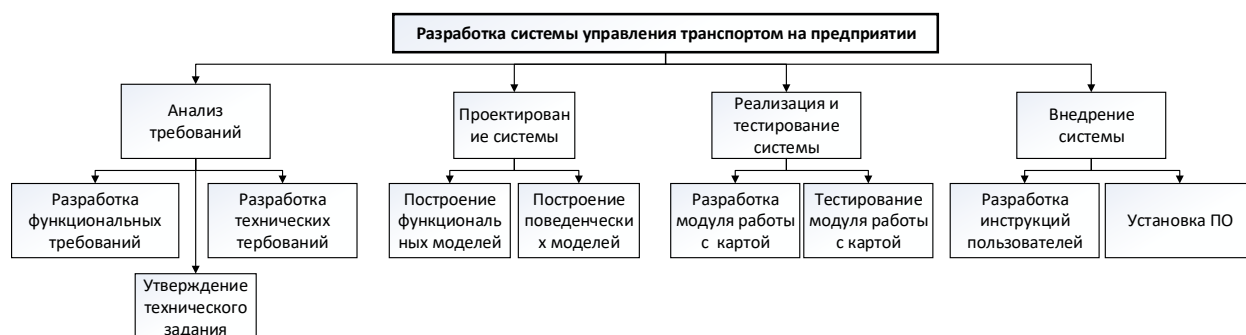


Рис. 6. Иерархическая структура работ

8. Подготовка сводного (укрупнённого) плана проекта. Данный план определяет последовательность этапов выполнения проекта без детализации конкретных работ и исполнителей. Данный план в дальнейшем может детализироваться и изменяться и используется для оценки общих сроков и трудоёмкости проекта.

9. Подготовка матрицы распределения ответственности. Матрица распределения ответственности (рис. 7) – связывает отдельные работы из WBS с ролями исполнителей из ORG. В матрице определяются основные исполнители для каждой работы проекта.

	Руководитель проекта	Аналитик	Программист	Системный администратор
Разработка функциональных требований	утверждает	выполняет	согласует	
Разработка технических требований	утверждает	выполняет		согласует
Утверждение технического задания	утверждает	выполняет	согласует	согласует
...				

Рис. 7. Матрица распределения ответственности

10. Разработка системы разбивки, соответствующей принятой в организации системе бухгалтерских счетов. На данном этапе необходимо определить, каким образом операции и работы проекта будут отражаться в бухгалтерских счетах.

11. Подготовка детального сетевого графика и плана использования ресурсов. Сетевой график (рис. 8) – граф показывающий зависимости между работами проекта и последовательность их выполнения для достижения конечной цели. Он используется для управления ресурсами и сроками выполнения

проекта. Структура ресурсов проекта (рис. 9) показывает потребность отдельных работ проекта в ресурсах и сроки их поставки. Структура стоимости проекта (рис 10) показывает распределение затрат проекта по этапам его выполнения.

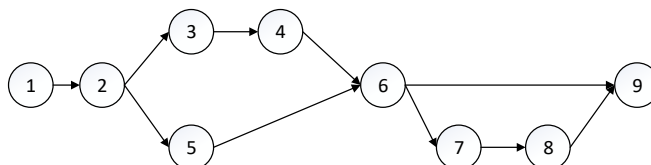


Рис. 8. Фрагмент сетевого графика проекта

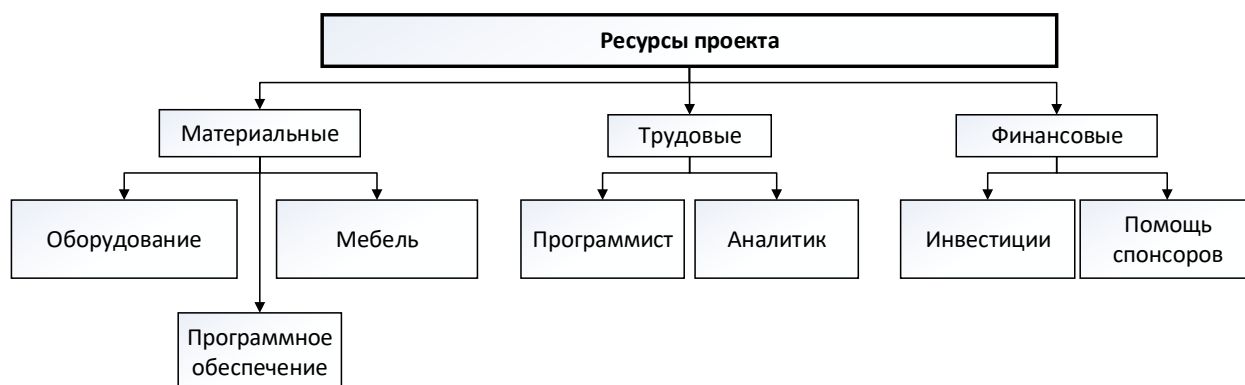


Рис. 9. Структура ресурсов проекта

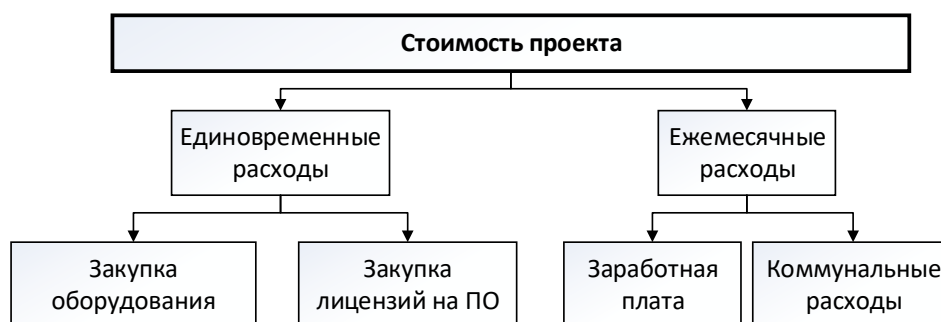


Рис 10. Структура стоимости проекта

12. Запуск системы наряд-заданий. На данном этапе необходимо в соответствии с сетевой моделью и матрицей ответственности раздать конкретные задания исполнителям проекта и определить сроки их выполнения.

13. Установка системы отчётности и контроля. На данном этапе необходимо определить перечень документов, оформляемых по результатам выполнения каждого этапа работ. Данные документы необходимы для контроля за ходом выполнения проекта.

Перечисленные этапы позволяют определить все необходимые элементы структуры проекта и определить порядок его реализации для достижения целей проекта в условиях имеющихся ресурсных ограничений компании. Таким образом структура проекта зависит от его целей и особенностей организации в которой данный проект реализуется. В тоже время структура проекта сама оказывает влияние на бизнес-процессы и организационную структуру компании.

1.4 Жизненный цикл проекта

Важным этапом структуризации проекта является определение этапов его жизненного цикла. Жизненный цикл проекта описывает последовательность этапов его выполнения, начиная с момента появления идеи проекта и до реализации продукции проекта. Классическая модель жизненного цикла проекта включает в себя четыре этапа (рис. 11).

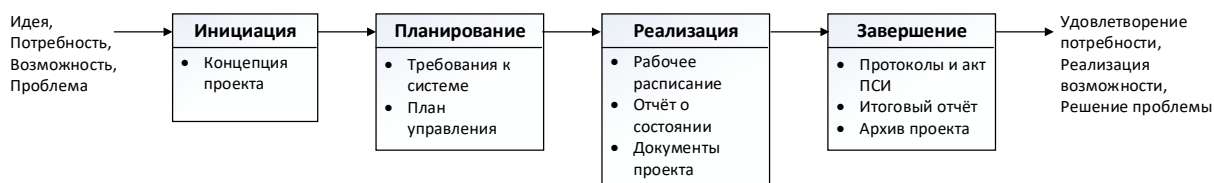


Рис. 11. Этапы жизненного цикла проекта

1. Этап инициации. На этапе инициации проекта происходит уточнение целей и задач проекта, а также определение продукции, которую мы хотим получить в процессе реализации проекта. На вход данного процесса поступает некоторая идея, проблема, требующая решения или потребность в некотором продукте. В ходе инициации данная проблема формализуется в конкретные цели и продукцию проекта. Основным документом, формируемым в процессе инициации, является концепция проекта, которая закрепляет задачи проекта, ресурсы и участников. Утверждение концепции проекта наделяет менеджера проекта полномочиями для управления ресурсами проекта, а также позволяет перейти к следующей стадии жизненного цикла.

2. Этап планирования. На этапе планирования происходит уточнение требований к системе и планирование реализации проекта. Основным документом, формируемым на данном этапе

является план управления проектом, который включает в себя иерархическую структуру работ, план управления содержанием, ресурсами, календарный план выполнения проекта, план управления рисками и изменениями.

3. Этап реализации. На этапе реализации происходит материализация планов и идей в форме документированного и протестированного программного продукта. На данном этапе составляется детальное рабочее расписание и осуществляется контроль и документирование процессов реализации продукта. Результатом данного этапа является законченный программный продукт.

4. Этап завершения. На этом этапе происходит сдача проекта заказчику и оформление итогового отчёта с результатами выполнения проекта.

Результатом выполнения проекта будет выступать удовлетворённая потребность, реализованная возможность или удовлетворённая потребность.

Модель жизненного цикла зависит от типа и особенностей проекта и может изменяться в реальных проектах. Кроме классической модели жизненного цикла есть гибкие модели, в которых используется спиральный подход к управлению проектом. При данном подходе этапы жизненного цикла повторяются с определённой периодичностью. Данный вариант жизненного цикла получил особую популярность в IT проектах, где требования зачастую чётко не формализованы и имеют склонность к частому изменению.

1.5 Организационная структура компании

Третьим компонентом структуризации проекта является выбор элементов организационной структуры компании. Организационная структура компании отражает ее внутреннее устройство, потоки управляющих воздействий, распределение труда и специфические особенности производства. Существуют следующие разновидности организационной структуры.

1. Функциональная организационная структура (рис. 12).

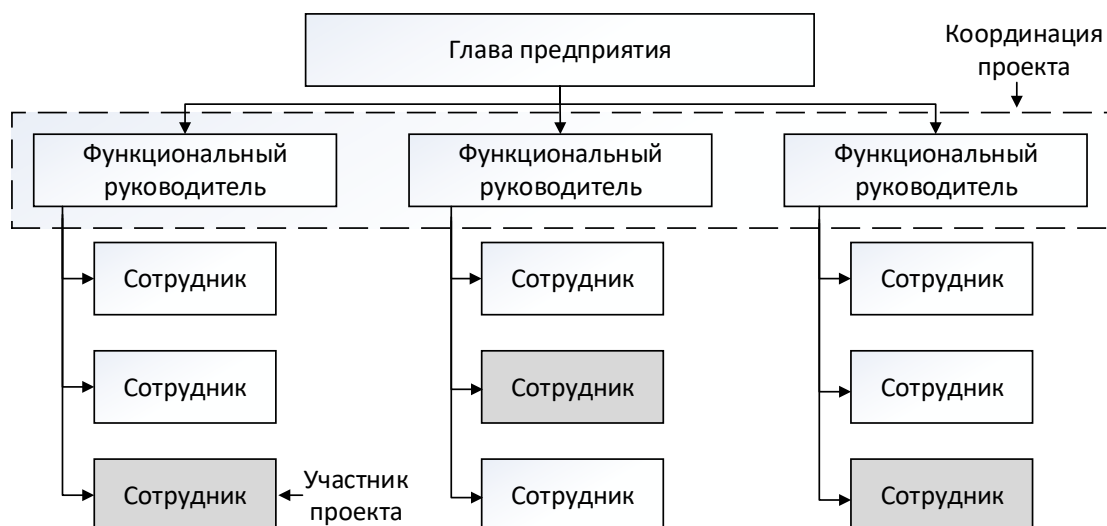


Рис. 12. Модель функциональной организационной структуры

По-другому данную структуру называют иерархической. Эта модель предполагает наличие чёткой вертикали управления. Каждый сотрудник работает в своём функциональном подразделении, имеет чётко сформулированные должностные обязанности и подчиняется только непосредственному руководителю функционального подразделения. Руководители функциональных подразделений подчиняются руководителю компании. Прямая коммуникация между сотрудниками разных функциональных подразделений отсутствует. Преимущества функциональной организационной структуры:

- Сохраняется принцип единоначалия. В результате система управления и подчинения становится прозрачной и однозначной.
- Понятные и стабильные условия работы. Каждый сотрудник работает в функциональном подразделении и выполняет строго определённый круг задач, которые не зависят от специфики реализуемых проектов.
- Такая организационная структура хорошо приспособлена для операционной деятельности.
- Специализация функциональных подразделений на отдельные виды задач позволяет накапливать опыт и экспертизу.

Недостатки функциональной модели организационной структуры:

- Затруднено принятие решений и коммуникация между сотрудниками различных подразделений. Любые коммуникации между сотрудниками могут осуществляться только через руководителей функциональных подразделений.

– Управление сконцентрировано и держится на компетенции высшего руководства в связи с чем высоки риски срыва проектов в случае выбытия кого-то из руководства.

– Не эффективен контроль за ходом выполнения проекта, т.к. нет целостной картины по проекту. Каждое подразделение отвечает за выполнение только своей части работ. При этом нет целостной картины хода проекта.

Функциональная организационная структура применяется в случае, если компания занимается реализацией только одного продукта, если реализуются сложные и длительные инновационные проекты. Также данная организационная структура применяется в научно-исследовательских организациях и крупных специализированных предприятиях.

2. Проектная организационная структура (рис. 13).

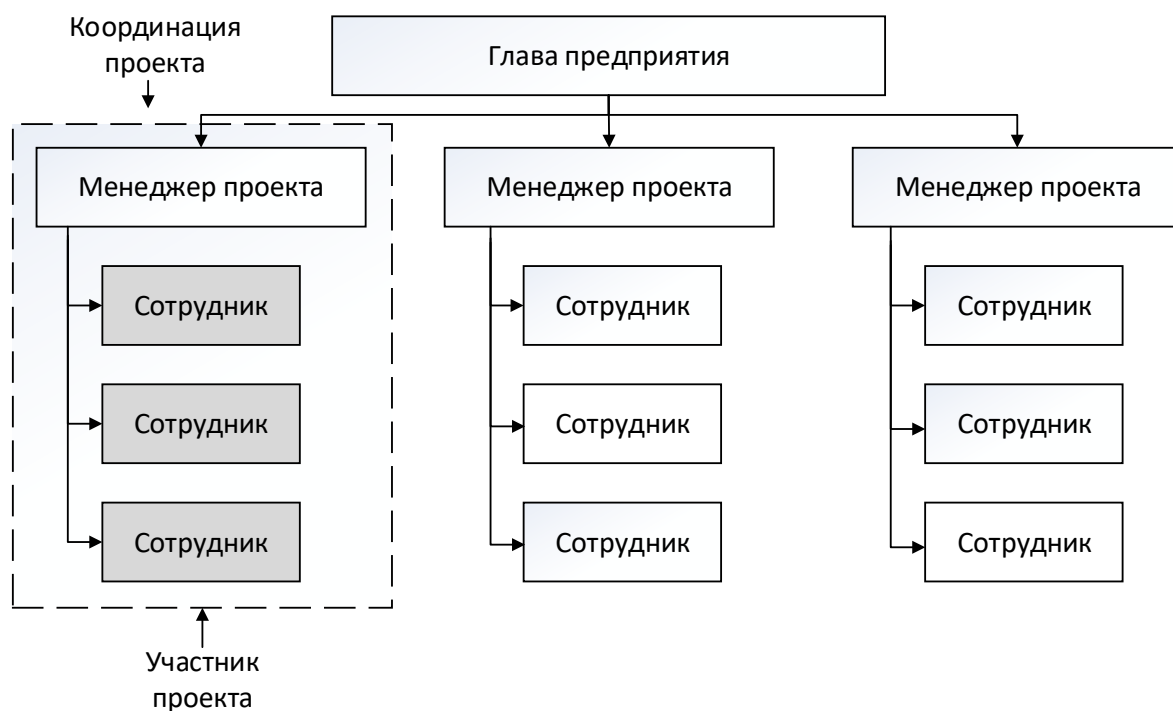


Рис. 13. Модель проектной организационной структуры

Проектная организационная структура – внутриорганизационная система, команда специалистов, выполняющая работы над одним проектом под управлением проектного менеджера. Данный вид организационной структуры характеризуется отсутствием чётко определённых подразделений и отделов. Сотрудники нанимаются в организации на период реализации проекта в соответствии с его спецификой. Обязанности

сотрудников определяются их ролью в конкретном проекте и могут меняться. После завершения проекта сотрудники переходят в новый проект на новые роли. Преимущества проектной организационной структуры:

- Адаптация организационной структуры под особенности проекта, позволяющая достигать максимальной эффективности процессов.

- Участие в реализации проекта специалистов, ориентированных на решение специфичных задач проекта.

- Простота коммуникации между участниками проекта. Все участники команды работают в одном подразделении и могут непосредственно взаимодействовать друг с другом.

- Полная вовлечённость сотрудников в реализацию проекта.

Недостатки проектной структуры:

- Медленный старт проекта. Это связано с необходимостью адаптации новых сотрудников в проектном коллективе и выработке организационных процессов.

- Опыт и практики управления отдельными проектами не аккумулируются и в каждом проекте вырабатываются заново.

- Проектные команды не сохраняются.

Проектная организационная структура применяется в организациях, чья деятельность носит проектный характер, компаниях одновременно реализующих несколько проектов в различных сферах деятельности, компаниях выполняющих краткосрочные и среднесрочные проекты по предварительным заказам.

3. Матричная организационная структура. Данный вид организационной структуры призван объединить преимущества двух предыдущих видов организационной структуры. При матричной организационной структуре все сотрудники работают в своих функциональных подразделениях как при функциональной структуре. Однако есть отдельная группа сотрудников – менеджеров проектов, которые занимаются непосредственным управлением отдельными проектами. Сотрудники функционального подразделения могут быть закреплены за конкретным проектом и будут выполнять работы в рамках этого проекта. Менеджер проекта призван упростить коммуникации между сотрудниками различных отделов. Матричная

организационная структура наиболее популярна у компаний, занимающихся разработкой программного обеспечения. Существует три разновидности матричной организационной структуры:

3.1. Слабая матрица (рис. 14). 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.

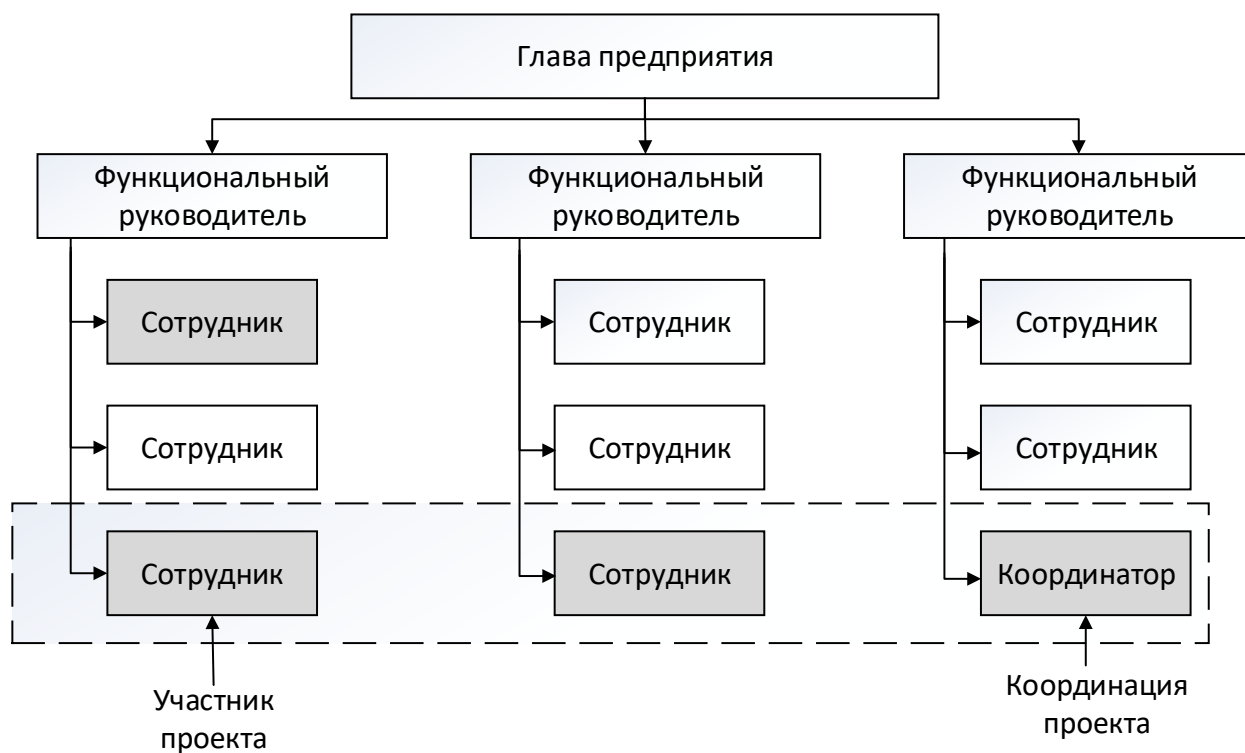


Рис. 14. Модель слабой матричной организационной структуры

3.2. Сбалансированная матрица (рис. 15). Данная организационная структура 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.

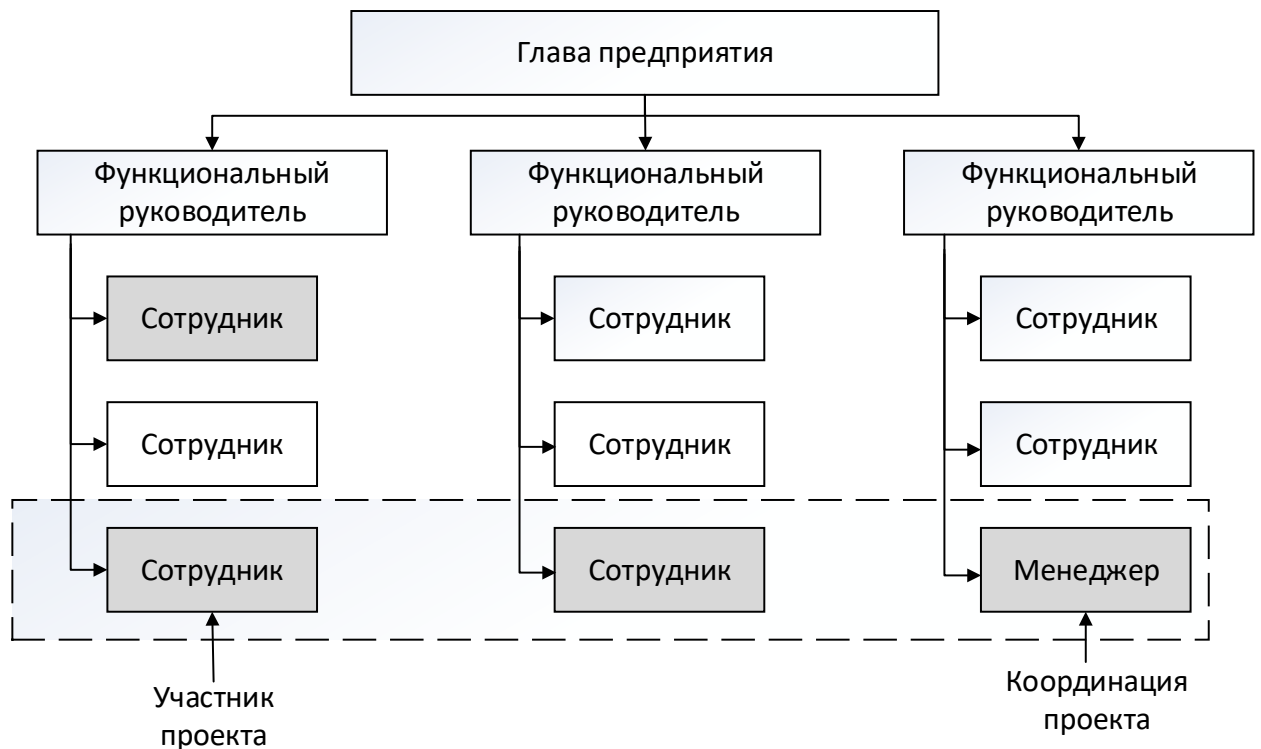


Рис. 15. Модель сбалансированной матричной организационной структуры

3.3. Сильная матрица (рис. 16). 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.

Преимущества матричной организационной структуры:

- Значительная активация деятельности руководителей и управляющих работников за счёт формирования проектных подразделений, взаимодействующих с функциональными.

– Разделение функций управления между функциональными и проектными руководителями.

– Упрощение коммуникации между сотрудниками различных функциональных подразделений, работающими над одним проектом.

Недостатки матричной организационной структуры:

– Нарушение принципа единоначалия. В результате возникают проблемы определения приоритетов задач поступающих от разных руководителей.

– Возникновение конфликтов между менеджерами функциональных подразделений и управляющими проектами.

– Трудности и длительность согласования при принятии решений.

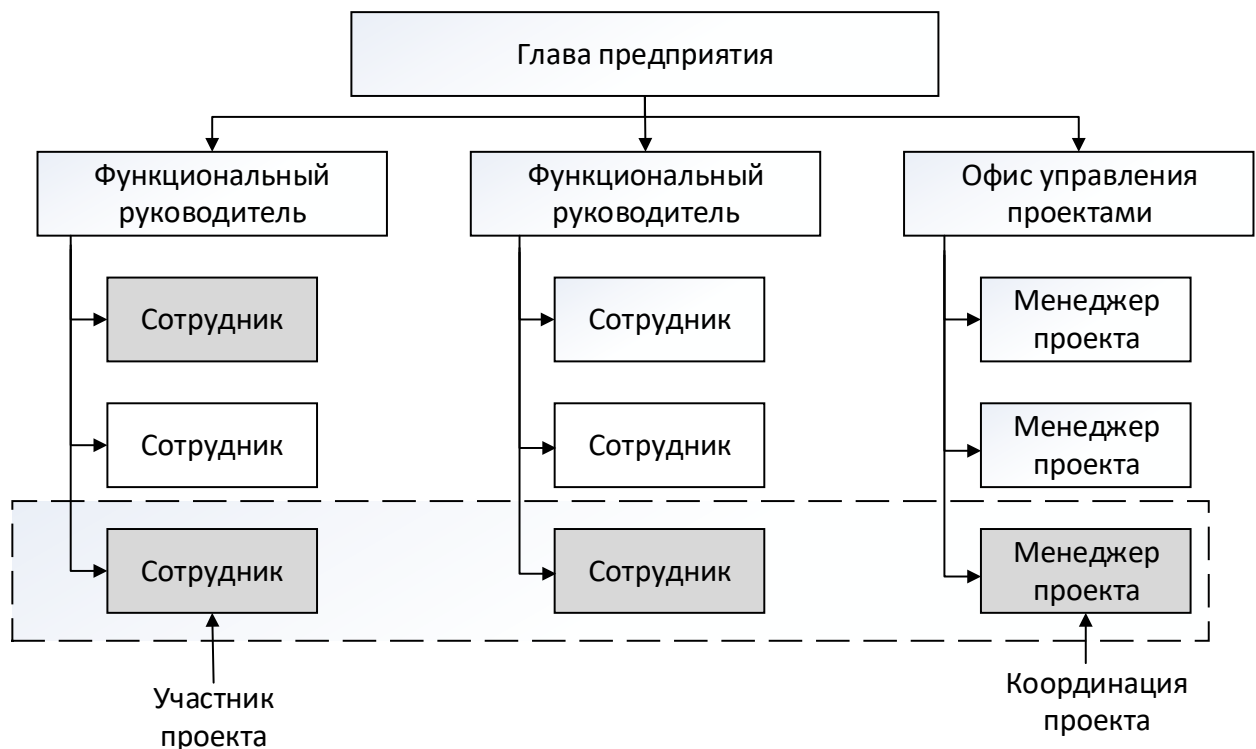


Рис. 16. Модель сильной матричной организационной структуры

Матричная организационная структура чаще всего применяется в организациях выполняющих множество типовых проектов в одно сфере деятельности.

Важно помнить, что организационная структура проекта – «живой» организм. Она начинает складываться на стадии планирования и может меняться в ходе проекта.

Подводя итог можно сказать, что структура проекта определяет его содержание, компоненты и порядок реализации. Также структура проекта влияет на организационную структуру компании и бизнес-процессы. Определение структуры проекта является одной из важнейших задач управления проектами.

1.6 История развития методов управления проектами

Становление управления проектами как самостоятельной отрасли научных исследований относится к концу XIX века. Однако сами методы управления проектами применялись человечеством значительно раньше. С задачей управления проектами человечество столкнулось со времён, когда начали реализовываться крупные и масштабные проекты. Создание египетских пирамид и другие подобные проекты требовали организации и координации большого количества работников. В результате вырабатывались наиболее эффективные практики организации работ, которые и выступали в качестве первых методов управления проектами. Однако зачастую эти правила не формализовывались в конкретные научные принципы и подходы и не было сложившейся практики их применения.

Принято считать, что основу современной теории и методологии управления проектами составляют лишь достижения прошлого столетия. В конце XIX века в некоторых странах начинают реализовываться крупномасштабные строительные проекты, требующие не только обработки огромного объёма сырья и материалов, но и организации работы тысяч рабочих, согласование работы десятков и сотен поставщиков и подрядчиков.

В начале XX века Фредерик Уинслоу Тейлор (1856–1915 гг.) проводит ряд многочисленных экспериментов, направленных на исследование трудовых процессов и производительности. Работы «Отца научного менеджмента» во многом стали основой целого ряда современных инструментов управления проектами.

Его ученик, Генри Гантт (1861–1919 гг.) концентрируется на изучении последовательности операций, выполняемых рабочими. Предложенные им аналитические диаграммы, включающие задачи,

отрезки задач и вехи, оказались крайне полезным и до сих пор актуальным средством управленческого анализа.

Основоположник «классической» школы управления, Анри Файоль (1841–1925 гг.) выделил пять функций менеджмента (предвидение, организация, распоряительство, координация, контроль), ставших концептуальной основой управления проектами.

Во время Второй мировой войны, в условиях жесточайшей экономии всех видов ресурсов и времени для выполнения огромного количества инженерных работ разрабатываются сложные методы сетевого планирования. В послевоенные годы эти методы получают широкое распространение в различных отраслях и сферах деятельности. Крупные промышленные корпорации начинают применение подобных методик для разработки новых видов продукции, модернизации производства и т. п.

В 1956 г. М. Р. Уолкер из корпорации «Du Pont de Nemours Co.» и Джон Келли из корпорации «Remington Rand» попытались использовать ЭВМ для составления планов-графиков крупных комплексов работ. В результате был создан рациональный и простой метод описания проекта с использованием ЭВМ (метод критического пути или CPM (Critical Path Method)).

Одним из наиболее известных проектов, на котором были впервые использованы методы моделирования и согласования комплекса работ, является проект разработки ракетной системы "Поларис", начатый в 1957 году. Данный проект имел жесткие ограничения по срокам, поскольку был привязан к предполагаемой дате ввода в эксплуатацию в СССР ракет, способных нести ядерные заряды и достигать территории США. В то же время в рамках данного проекта необходимо было разработать, провести сборку и тестирование значительного количества не имеющих аналогов компонент. Реализация проекта, объединявшего около 3800 основных подрядчиков и состоявшего из 60 тысяч задач, была поручена Главному управлению вооружений ВМС США. В целях управления реализацией этого проекта корпорацией "Локхид" и консалтинговой фирмой "Буз, Аллен энд Гамильтон" был создан специальный метод планирования работ на основании оптимальной логической схемы процесса, названный методом анализа и оценки программ PERT (Program Evaluation and Review Technique). Использование метода PERT позволило руководству программы

точно знать, что требуется делать в каждый момент времени и кто именно должен это делать, а также вероятность своевременного завершения отдельных операций. Руководство программой оказалось настолько успешным, что проект удалось завершить раньше запланированного срока. Благодаря такому успешному началу данный метод управления был засекречен и вскоре стал использоваться для планирования проектов во всех вооруженных силах США. Методика отлично себя зарекомендовала при координации работ, выполняемых различными подрядчиками в рамках крупных проектов по разработке новых видов вооружения.

В зарубежной литературе основателем теории и методологии управления проектами называют известного проектного руководителя из немецкой фирмы «Dornier» Роланд Гутча. В 1937 году он подготовил первую разработку по матричной организации для руководства и осуществления сложных проектов, а в 1965 году организовал Международную ассоциацию управления проектами «ИНТЕРНЕТ», объединяющую более двадцати крупных национальных ассоциаций в области управления проектами.

Управление проектом в СССР зародилось в 20–30-е годы в период индустриализации. Это время начала реализации целого ряда беспрецедентных по масштабу проектов (ГОЭЛРО, Днепрогэс, создание больших территориально-индустриальных комплексов и т. п.). Первый опыт промышленного строительства, требующий очень высокого уровня планирования и организации, является основой теории потока и современной научной организации труда и управления производством. Огромный вклад в развитие теории потока внесли А. В. Барановский, О. А. Вутке, М. В. Вавилов, А. А. Гармаш, Б. П. Горбушин, Н. И. Пентковский и др.

Активное развитие и внедрение методов сетевого планирования и управления начинается в конце 50-х годов. Первые работы по сетевым методам были опубликованы М. Л. Разу, С. И. Зуховицким, И. А. Радчиком. Уже в конце 60-х годов в СССР были разработаны уникальные сетевые модели, значительно более гибкие и мощные, чем зарубежные аналоги (CPM и PERT). К 1975 году эти методы применялись уже на более чем 15 % строек в стране.

Развитие сетевого планирования активизировалось с появлением специальных научно-исследовательских институтов, с

началом преподавания передовых методов в высших учебных заведениях СССР.

В 80-е годы усовершенствованные методы сетевого планирования и управления проектами переходят на качественно иной уровень использования — в составе автоматизированных систем управления. Первыми программными комплексами для управления проектами становятся «Калибровка-2» и «А-План». В это же время в Московском институте управления (ныне — Государственный Университет Управления) открывается первая в стране кафедра автоматизированных систем управления строительством, активно разрабатывающая программные продукты по управлению строительными проектами. Специалистами кафедры (О. В. Козловой, М. Л. Разу, Г. А. Брянским, О. А. Овсянниковым и другими) разработаны такие инструменты управления проектами, как сетевые матрицы, информационно-технологические модели, матрицы разделения административных задач управления и многие другие.

В 1991 года российская ассоциация управления проектами (СОВНЕТ) вступает в международную ассоциацию «ИНТЕРНЕТ». Россия становится полноправным членом международного сообщества проектного управления.

Сегодня инструментарий управления проектами используется во всех сферах целенаправленной деятельности. Во многих развитых странах органы власти используют средства управления проектами в повседневной деятельности не только для реализации крупных государственных проектах, но и для организации эффективного функционирования собственного аппарата. Использование технологий управление проектами становится обязательным при реализации инвестиционных и строительных проектов.

Контрольные вопросы

1. Дайте определение понятия проект и управление проектами.
2. Какие из перечисленных видов деятельности относятся к проектной: внедрение новой процедуры подбора персонала, выпуск плановой партии продукции, проведение рекламной компании,

строительство офисного здания, подготовка квартального отчёта, разработка программного обеспечения, поддержка пользователей?

3. Перечислите особенности проекта.

4. Классифицируйте приведённый проект по различным основаниям: Проект расширения ассортимента розничной торговли компании Amazon.com за счёт продажи и доставки различных видов продукции по всему миру.

5. Из каких компонентов состоит структура проекта?

6. Охарактеризуйте этапы жизненного цикла проекта.

7. Перечислите особенности функциональной организационной структуры компании.

8. Перечислите особенности проектной организационной структуры компании.

9. Перечислите разновидности и особенности матричной организационной структуры компании.

10. Охарактеризуйте основные этапы развития управления проектами за рубежом и в России.

2 Фаза инициации проекта

2.1 Работы, выполняемые на фазе инициации проекта

Инициация проекта – это стадия проекта, на которой выполняется ряд работ обеспечивающих успешный запуск проекта. Эффективность процессов, выполняемых на данной стадии, минимум на пятьдесят процентов определяет успешность самого проекта. На данной стадии происходит определение целей и задач проекта, определение целесообразности реализации проекта, назначение руководителя проекта, разработка устава проекта, идентификация участников проекта и заинтересованных лиц.

Очень часто данному этапу уделяется не достаточное внимание при управлении проектами. Такой подход приводит к большому количеству проблем на последующих этапах реализации проекта и может привести к его провалу. Во время инициации необходимо понять точные цели проекта и определить возможность и целесообразность его реализации для организации.

Другими словами инициация проект – это момент когда ещё есть возможность проанализировать все плюсы и риски реализации проекта и принять решение делать ли проект вообще. На данном этапе ещё не затрачены ресурсы компании. В это время можно отказаться от реализации проекта, если руководитель видит слишком высокий уровень рисков данного проекта. Такое решение не нанесёт никаких последствий на репутацию компании и менеджера проекта. Если же концепция такого проекта будет утверждена и будет назначен руководитель проекта, то последствия от неуспешного выполнения проекта будут значительно большими.

Таким образом, главная задача этапа инициации проекта это принятие решения о том будет ли компания реализовывать его или нет. Для принятия данного решения необходимо чётко сформулировать и описать основную информацию о проекте, сделать обоснование его целесообразности, описать конечный продукт и результаты проекта, определить, какие ресурсы потребуются для его реализации. Таким образом, необходимо формализовать всю информацию о проекте, которая позволит принять решение о целесообразности выполнения данного проекта.

2.2 Модель видения проекта

Одним из инструментов, применяемых для формализации данных о проекте на этапе инициации, является модель видения проекта, разрабатываемая в соответствии с методологией Rational Unified Process (RUP). Данная методология была создана Филиппом Крачтенем (Philippe Kruchten), Иваром Якобсоном (Ivar Jacobson) и другими сотрудниками компании «Rational Software» в качестве дополнения к языку моделирования UML. RUP опирается на проверенные практикой методы анализа, проектирования и разработки программного обеспечения, методы управления проектами. В основе методологии RUP лежат следующие основные принципы:

- 1 ранняя идентификация и непрерывное (до окончания проекта) устранение основных рисков;
- 2 концентрация на выполнении требований заказчиков к исполняемой программе (анализ и построение модели прецедентов (вариантов использования));
- 3 ожидание изменений в требованиях, проектных решениях и реализации в процессе разработки;
- 4 компонентная архитектура, реализуемая и тестируемая на ранних стадиях проекта;
- 5 постоянное обеспечение качества на всех этапах разработки проекта (продукта);
- 6 работа над проектом в сплочённой команде, ключевая роль в которой принадлежит архитекторам.

Модель видения проекта содержит начальное описание будущего проекта и обоснование полезности и целесообразности его разработки. В нём кратко описываются цели и задачи проекта, основные продукты, заинтересованные стороны и обоснование полезности проекта. Видение необходимо для формирования единого представления о будущем проекте у всех заинтересованных сторон, а также для принятия решения о дальнейшей реализации проекта. Создание видения проекта зачастую предшествует разработке концепции проекта, как основного документа, определяющего его содержание.

Видение проекта в концепции RUP строится, как документ, последовательно детализирующий потребителей, их потребности, цели и границы проекта, таким образом, чтобы читатель, не

знакомый с предметной областью, читая документ, постепенно погружался в нее. Поэтому каждый последующий пункт детализирует информацию предыдущих пунктов.

Видение программного продукта предполагает ответ на пять основных вопросов:

- кто будет приобретать и использовать продукт (целевая аудитория);
- какие нужды пользователей продукт удовлетворяет;
- каковы критические характеристики продукта, чтобы удовлетворить эти нужды и сделать продукт успешным;
- чем продукт похож на существующие аналоги и чем он от них отличается;
- какой предполагается срок и бюджет для создания продукта.

Видение проекта представляет собой документ, имеющий следующую структуру.

1. Введение. Во введении описываются цель документа, его контекст, основные определения, акронимы и сокращения, даются ссылки на другие документы и характеризуется содержание документа. Задача этого пункта – дать читателю документа возможность понять, что описано в этом документе и для чего предназначена описываемая система. Основные элементы введения:

- Цель документа – формальная цель разработки документа. Обычно, целью документа объявляется сбор, анализ и определение высокоуровневых потребностей заинтересованных лиц и конечных пользователей системы, а также характеристик системы необходимых для удовлетворения этих потребностей. Детали того, как система отвечает данным потребностям, записываются в модели прецедентов и, возможно, других документах спецификации.

- Область действия документа – краткое описание границ этого документа. Указывается продукт или приложение, которое будет создаваться или изменяться. Формулируется цель приложения, версии и новые предоставляемые функции в развитие существующей системы.

- Определения, аббревиатуры и сокращения – перечень всех необходимых для понимания сокращений либо ссылка на отдельный документ глоссария.

– Ссылки – в этом разделе приводится полный список источников, на которые существуют ссылки в Видении или к которым можно обратиться за справкой. Информация может быть представлена ссылкой на приложение к видению или ссылкой другой документ.

– Обзор документа – здесь описывается, остальное содержимое Видения и способ структурирования информации в нем. Целью данного раздела является дать читателю возможность быстро сориентироваться в структуре документа.

2. Позиционирование. Раздел «позиционирование» содержит определение решаемой проблемы, указывается целевой заказчик продукта и исследуются преимущества изделия перед аналогами на рынке. Этот раздел не используется для формулировки конкретных требований – в нем только содержатся обоснования, почему возникла необходимость разработки данного продукта и почему были выдвинуты те или иные требования. Основные элементы данного раздела:

– Необходимость в продукте. В данном пункте кратко описывается: почему данный продукт будет востребован потребителем и как возникла возможность для разработки продукта.

– Постановка проблемы. В данном пункте описываются проблемы, вставшие перед пользователями, которые может решить разрабатываемый продукт. Явная и четкая формулировка проблем пользователя решаемых продуктом, помогает гарантировать наличие потребителя у создаваемого продукта, обеспечить достаточное число потребителей, избежать реализации не нужного функционала, реализовать весь необходимый функционал. Для каждой проблемы указываются: формулировка проблемы, кого она затрагивает, как она появилась и как она появилась.

– Формулировка позиции продукта. В данном пункте описывается целевая аудитория, разрабатываемый продукт, его конкуренты и ключевые отличия от них. В данном пункте проводится обобщенный анализ конкурентных преимуществ нового решения и дается общее определение на высоком уровне абстракции особого положения, которое продукт должен занять в линейке решений для данной предметной области. Для каждой целевой группы указывается следующая информация: группа

потребителей, описание задач, которые эта группа будет решать, категория нового продукта, ключевая выгода, перечисление основных конкурентов, список новых возможностей и характеристик, которые продукт предоставляет данной группе потребителей.

3. Описание заинтересованных сторон и пользователей. Данный раздел содержит подробную информацию о целевой аудитории продукта, его потребителях и окружении. Нечёткое выявление потребителей и их требований к продукту зачастую приводит к тому, что даже у качественно сделанного программного обеспечения будет не достаточная аудитория или продукт будет покрывать только часть нужд пользователей. Поэтому в данном разделе важно внимательно составить список заинтересованных лиц, перечислив их всех, и сформулировать их потребности. Данный раздел содержит следующие элементы:

- Состояние рынка. В данном пункте описываются ключевые показатели рынка сбыта, важные для разработки продукта. При этом необходимо оценить размер рынка и его потенциальный рост, указав число возможных пользователей, либо возможного дохода, например на основе оценки текущих расходов пользователей на решение задач, которые решает новый продукт. Кроме того, следует соотнести новый продукт с целями самой организации разработчика (особенно это важно при разработке для внутренних заказчиков). Для этого надо описать: как продукт связан с использованием новых технологий, как повлияет выход нового продукта на репутацию компании, насколько продукт соответствует стратегическим целям компании.

- Описание заинтересованных лиц. В данном пункте перечисляются все заинтересованные лица, которые будут использовать систему или с которыми она будет связана. Заинтересованным в проекте является лицо, удовлетворение интересов которого повышает ценность продукта. При этом под понятием «заинтересованного лица» понимается не конкретные люди, работающие с системой, а функциональные роли, которые они играют в этот момент. Для каждого заинтересованного лица, необходимо указать роль в проекте, потребности и их важность, критерий успеха удовлетворения каждой потребности, текущие и предлагаемые способы удовлетворения потребностей. Можно выделить три категории заинтересованных лиц:

а. Первая категория заинтересованных лиц – это пользователи системы. Их потребности легко учесть, поскольку они будут непосредственно привлекаться к определению и использованию системы.

б. Вторую категорию составляют не прямые пользователи, а также те, на кого воздействуют только последствия разработки. Этих заинтересованных лиц можно найти среди компаний, занятых в сфере деятельности, связанной с разрабатываемым приложением. Например, это могут быть инвесторы проекта, владельцы организации или субподрядчики по разработке системы.

в. Кроме людей, также необходимо учесть интересы сопряженных технических систем и внешних организаций, например государственных структур. Такой список составляется для обеспечения учета ключевых интересов заинтересованных лиц при планировании функциональности продукта.

– Описание пользователей. В данном пункте перечисляются все идентифицированные на данный момент пользователи системы. Особое внимание уделяется пользователям, являющимся заинтересованными лицами. Для каждого пользователя, не являющегося заинтересованным лицом, указывается: заинтересованное лицо, представляющее его интересы, потребности и их важность, критерий успеха удовлетворения каждой потребности, текущие и предлагаемые способы удовлетворения потребностей.

– Окружение пользователя. В данном пункте содержится описание рабочего окружения конечных пользователей, характеристики организации, в которой будет использоваться продукт, условия работы продукта.

– Альтернативы и конкуренты. В данном пункте перечисляются все конкуренты разрабатываемого продукта. Не существует программных продуктов, которые бы вообще не имели конкурентов. В качестве альтернативы всегда можно рассматривать вариант ручного решения проблемы, без использования программных средств, создание самодельного программного решения вместо приобретения разрабатываемого продукта или отказ от решения проблемы. Для анализа альтернатив и конкурентов можно создать таблицу со следующими данными: заинтересованное лицо, приоритет потребности, текущее решение

потребности, его плюсы и минусы, предлагаемое решение потребности, его плюсы и минусы.

4. Обзор продукта. Данный раздел содержит краткий обзор изделия, описание его перспектив и ключевых возможностей в среде применения, предположения и зависимости, оценка стоимости разработки (калькуляция), вопросы лицензирования и инсталляции. Данный раздел содержит следующие элементы:

- Формула продукта – предлагается общее определение на высоком уровне абстракции особого положения, которое продукт должен занять в линейке решений для данной предметной области.

- Перспективы продукта – указываются планы развития продукта. Прогнозируется время жизни продукта.

- Набор основных возможностей – перечисляются основные выгоды заинтересованных лиц, а также характеристики системы, которые обеспечивают эти выгоды. Функции должны описываться на естественном языке пользователя. Эти функции составляют основу для определения продукта, а также управления проектом в целом. Впоследствии эти возможности будут раскрываться в техническом задании в функциональные требования к системе.

- Предположения и зависимости – перечисляются все факторы, влияющие на характеристики системы, указанные в данном документе. Перечисляются все предположения, которые в случае изменения, приведут к изменению данного документа. А также перечисляются связанные технические системы для выявления необходимости сопряжения с ними.

- Стоимость и цена продукта – указывается предполагаемая стоимость разработки и ценовая категория, по которой продукт будет поставляться конечному потребителю.

- Лицензирование и установка – кратко описываются характеристики процедуры установки продукта и требования по его лицензированию.

5. Характеристики продукта. Характеристики продукта это высокоуровневые возможности, необходимые для удовлетворения потребностей пользователей продукта. Для каждой характеристики должен быть указан ее приоритет, статус (например, предложена, утверждена), сложность, риск.

6. Ограничения для продукта. Данный раздел содержит список ограничений и требований к разрабатываемому продукту. Основные элементы данного раздела:

- Требования к качеству – данный пункт определяет требования к производительности, надежности, удобству использования и другим характеристикам качества программного продукта.

- Применяемые стандарты – здесь перечисляются стандарты, которым должен соответствовать продукт: коммуникационные стандарты, стандарты совместимости платформ, а также стандарты качества и безопасности.

- Системные требования – определяют все системные требования, необходимые для поддержки приложения, операционные системы и сетевые платформы, конфигурации, память.

- Требования к производительности. В данном пункте перечисляются основные вопросы, связанные с производительностью разрабатываемого продукта. К ним относятся: фактор нагрузки, создаваемой пользователями, надежность, время ответа системы.

- Требования к документации – здесь описывается, какую документацию необходимо разработать для поддержки успешного внедрения и эксплуатации продукта.

Таким образом, видение проекта является документом, определяющим цели проекта, фиксирующим его границы, четко определяющим пользователей системы и способы удовлетворения их ключевых потребностей, позволяющим позиционировать продукт на рынке, осуществлять долгосрочное планирование и контролировать объем реализуемой функциональности. Данный документ является отправной точкой в управлении проектом и служит для принятия решения о целесообразности его реализации.

2.3 Пример видения программного продукта

Рассмотрим пример видения продукта для проекта разработки сервиса по поиску схожей по звучанию музыки.

Введение.

Целью данного документа анализ потребностей, выявление заинтересованных лиц и закрепление характеристик и границ предлагаемого программного продукта.

В данном документе подробно будут рассмотрены цель и задачи, решаемые в рамках предлагаемого проекта, описаны заинтересованные стороны, собраны основные требования и границы, поставленные перед проектом.

В данном документе будут использоваться следующие обозначения:

- ИНС искусственная нейронная сеть (самообучающийся алгоритм), для определения степени схожести музыкальных произведений по их звучанию;
- МС – музыкальный сервис с возможностью поиска похожих музыкальных композиций.

Позиционирование продукта.

В современных музыкальных сервисах можно получить рекомендации по жанру, исполнителям, году выпуска, но отсутствует возможность автоматически найти композиции, просто похожие по звучанию. Данный программный продукт позволит решить эту задачу. Потребность в данном функционале была выявлена работниками нашей компании, на основе своего опыта, а также данных опроса людей в социальных сетях. МС, реализующий данный функционал, является внутренней инициативой компании.

Описание заинтересованных сторон.

Использовать предлагаемый продукт сможет любой любитель музыки по всему миру. Даже пользователи конкурентов могут выбрать этот продукт из-за расширенного функционала. Использовать эту систему сможет любой человек, имеющий доступ к компьютеру, смартфону, планшету и т. п., подключенному к сети интернет.

Обзор продукта.

Разрабатываемый программный продукт представляет собой кросс платформенное приложение с веб-интерфейсом. Основные функции:

- прослушивание музыки;
- сортировка и выбор композиций по какому-либо тегу;
- поиск композиций, схожих по звучанию.

В перспективе проект позволит:

- занять значительную долю рынка музыкальных сервисов;
- значительно повысить имидж и значимость компании-разработчика;
- качественно улучшить навыки персонала компании в сфере применения новейших технологий, таких как — нейронные сети, разработки мульти платформенных систем.

Характеристики продукта.

Количество ошибок, совершаемых системой при поиске должно быть не более 5%

Время поиска треков должно составлять не более 15 секунд.

Доступность сервиса на различных платформах 24/7.

Ограничения для продукта.

Для предоставления музыкальных материалов необходимо договориться с их правообладателями о размещении материалов в нашем продукте, с возможностью их прослушивания пользователями.

С приобретенной подпиской пользователь получит доступ к обширной коллекции видео и аудио материалов, возможность создавать свои плейлисты и прослушать плейлисты других пользователей, получить уникальные рекомендации, основанные на понравившихся материалах.

В перспективе приложение позволит пользователям создавать личные подкасты, для трансляции интересной, любимой музыки.

Для реализации данного функционала потребуются квалифицированные специалисты в области проектирования и обучения глубоких нейронных сетей, а также использование сопутствующих технологий и фреймворков. Использование различных платформ предполагает наличие необходимых навыков и опыта у команды разработчиков.

В связи с ресурсоемкостью обучения ИНС необходимо позаботиться о достаточности вычислительных мощностей.

Для разработки и тестирования мультиплатформенного МС необходимо наличие соответствующих программных средств и оборудования.

Учитывая то, что аудитория пользователей включает в себя множество различных категорий, интерфейс должен быть создан с применением техник UX. Разные языковые категории должны получить качественную и полную локализацию.

Качество МС можно оценить по следующим параметрам:

- итоговый процент ошибки алгоритма поиска на реальных данных;
- успешное прохождение всех заявленных тестов;
- отказоустойчивость;
- положительное мнение тестовой группы пользователей на UI.

Предполагается, что разрабатываемая система будет обладать высокой отказоустойчивостью за счет ввода в эксплуатацию дополнительных резервных серверов. Данные сервера будут использоваться при высокой нагрузке на основные сервера, а также в случае их выхода из строя.

Доступ к полному функционалу реализуемого проекта будет осуществляться с помощью оформления пользователями сервиса ежемесячных подписок. Пользователи, у которых не оформлена подписка на сервис, будут получать рекламу. Таким образом, при успешной маркетинговой компании и большом количестве пользователей, обеспечивается экономическая эффективность.

2.4 Структура концепции проекта

Основным документом, создаваемым на этапе инициации проекта является концепция проекта. Концепция проекта содержит информацию о целях и задач проекта, его структуре, ресурсах, необходимых для его реализации, а также ожидаемых результатах и критериях приёмки продукта проекта. Основное назначение данного документа – создание единого видения проекта всеми участниками и заинтересованными сторонами.

Также в концепции проекта закрепляется его руководитель, если он не был утверждён раньше. Утверждение концепции проекта даёт руководителю проекта полномочия распоряжаться ресурсами компании, выделенными на выполнение данного проекта.

Типовая структура концепции проекта содержит следующие разделы:

1. Название проекта;
2. Цели проекта;
3. Результаты проекта;
4. Допущения и ограничения;

5. Ключевые участники и заинтересованные стороны;
6. Ресурсы проекта;
7. Сроки;
8. Риски;
9. Критерии приемки;
10. Обоснование полезности проекта.

Рассмотрим содержание выделенных разделов подробнее.

1. Название проекта содержит краткое наименование проекта, используемое в компании для его идентификации. Название должно быть кратким, чётким и однозначным. Из названия проекта должна быть понятна его суть и назначение. Название должно выделять проект среди множества других проектов, реализуемых в компании и подчёркивать его уникальность.

2. Цели проекта. Цели проекта показывают основное назначение данного проекта для компании разработчика и основных потребителей программного продукта. Они должны описывать текущие бизнес-потребности и согласовываться со стратегическими целями компании. Цели должны быть значимыми, конкретными, измеримыми, реальными. Можно выделить следующие виды целей, которые могут достигаться в рамках реализации проекта:

- удовлетворение существующей потребности на рынке;
- реализация стратегических целей компании. Это может быть завоевание доли рынка, повышение объёмов продаж и прибыли, снижение издержек и расходов;
- изменения в компании. Данная цель будет у проектов внутренней автоматизации и внедрения информационных технологий в деятельность собственной компании с целью повышения эффективности бизнес процессов или снижения издержек;
- выполнение контрактов. Данный вид целей будет в случае разработки программного обеспечения под заказ. Эти цели проекта будут пересекаться с целями заказчика;
- решение специфических проблем. Такая цель будет у проектов доработки программного обеспечения и исправления найденных ошибок в существующих программах.

У проекта всегда должна быть только одна главная цель, на достижение которой он направлен. Главная цель может включать в

себя ряд подцелей и задач, которые надо решить для достижения главной цели. Эти подцели также должны описываться в данном разделе концепции. Важной особенностью формулировки целей проекта является их измеримость. Должна быть возможность количественно измерить и оценить степень достижения поставленной цели по итогам реализации проекта.

3. Результаты проекта. Показывают, что конкретно будет получено после завершения проекта. Данный раздел концепции проекта может включать в себя следующие виды результатов:

- бизнес выгоды, которые получит заказчик в результате реализации проекта;
- продукт или услуга, которые будут произведены в рамках реализации проекта;
- высокоуровневые требования – ключевые характеристики произведённого продукта проекта, важные для заказчика.

Также как и цели, результаты проекта должны быть количественно измеримыми для подтверждения получения описанных результатов при реализации проекта.

4. Допущения и ограничения проекта. Данный раздел содержит описание начальных допущений и ограничений, которые применяются в реализуемом проекте. Допущения включают в себя набор условий и предположений, которые принимаются в рамках данного проекта. Такие предположения связаны с возможными рисками проекта и позволяют снизить последствия от их реализации и переложить их на заказчика. В качестве примера допущений может являться предположение о том, что стоимость лицензий на ПО не изменится во время реализации проекта или курс доллара существенно не изменится по ходу реализации проекта.

Ограничения определяют некоторые условия и рамки, в которых будет происходить реализация проекта. В проекте могут быть следующие категории требований:

- нормативные ограничения. Определяют требования к организации процессов разработки и сертификации программного продукта;
- технические ограничения. Определяют особенности аппаратного и программного обеспечения разрабатываемой системы;

- ограничения безопасности. Определяют специфические требования к защите информации в системе.

Также в данном разделе должны быть прописаны те функциональные требования, которые могут подразумеваться заказчиком, но не являются предметом реализации данного проекта.

5. Ключевые участники и заинтересованные стороны. В данном разделе описываются все участники и заинтересованные стороны проекта. К участникам проекта относятся лица, непосредственно участвующие в реализации проекта и оказывающие влияние на него. К заинтересованным сторонам относятся те лица организации, которые напрямую не участвуют в реализации проекта, но их интересы будут затронуты в рамках реализации проекта или после его завершения. Для описания заинтересованных сторон проекта в зарубежной литературе используется термин стейкхолдеры [2]. Ключевыми участниками проектов разработки программного обеспечения являются следующие лица и организации:

- спонсор проекта. Лицо или группа лиц, финансирующих выполнение проекта. Следует учитывать тот факт, что спонсор проекта в большинстве случаев будет совпадать с заказчиком проекта. Однако это не обязательно;

- заказчик проекта. Лицо или организация, которая будет использовать результаты проекта. Заказчик проекта не всегда будет являться непосредственным пользователем проекта, а может выступать в качестве посредника между разработчиками и пользователями;

- пользователи результатов проекта. Лица, непосредственно использующие разработанное в рамках проекта программное обеспечение;

- куратор проекта. Руководитель организации разработчика проекта уполномоченный принимать решения о выделении ресурсов на реализацию проекта, а также возможность внесения в него изменений;

- руководитель проекта. Представитель исполнителя, ответственный за реализацию проекта в полном объёме и в срок и уполномоченный распоряжаться ресурсами, выделенными на реализацию проекта;

– команда исполнителей. Сотрудники организации исполнителя, непосредственно задействованные в реализации проекта;

– соисполнители проекта. Это внешние субподрядчики и поставщики ресурсов.

6. Ресурсы проекта. В данном разделе перечисляются все ресурсы, которые необходимы на реализацию данного проекта. Выделяют следующие категории ресурсов:

- человеческие ресурсы и требования к их квалификации;
- материальные ресурсы;
- лицензии на средства разработки и тестирования;
- финансовые ресурсы.

Спецификой проектов по разработке программного обеспечения является тот факт, что основными ресурсами, необходимыми для их реализации и влияющими на их стоимость являются человеческие ресурсы. Необходимость в человеческих ресурсах определяется трудоёмкостью проекта и сроками его реализации. Оценка трудоёмкости проекта разработки программного обеспечения является сложной задачей и требует учёта большого числа особенностей и характеристик проекта. Т.к. на этапе инициации не достаточно данных для точной оценки трудозатрат, то хорошей будет считаться оценка трудоёмкости в диапазоне от -50% до +100%. Для оценки трудоёмкости проекта на этапе инициации можно использовать предположительный объём программного кода. Объём кода измеряется в тысячах строк кода KSLOC. Он зависит от выбранного языка программирования и применяемых технологий разработки. Объём кода для новых проектов может быть вычислен на основе анализа данных о реализации похожих проектов. Далее объём кода предполагаемой программы делится на среднюю производительность программистов в час. Полученная величина и будет начальной оценкой трудоёмкости реализации программного обеспечения в человеко-часах. Полученная оценка трудоёмкости показывает лишь затраты, непосредственно связанные с реализацией программного обеспечения. В рамках проекта будет выполняться множество других работ. Считается, что непосредственное кодирование программного обеспечения составляет одну четвёртую от всех работ по проекту. Поэтому рассчитанную трудоёмкость

необходимо ещё умножить на 4, чтобы получить оценку трудоёмкости реализации всего проекта в целом. Получим следующую формулу для оценки трудозатрат T .

$$T = \frac{4V}{p}, \text{ чел. – мес.}$$

где: V – объём программного кода, KSLOC

p – средняя производительность программиста, KSLOC/мес.

Имея информацию о трудоёмкости проекта можно определить оптимальное количество исполнителей проекта и сроки его реализации.

К материальным ресурсам относятся компьютеры, на которых будет непосредственно производиться разработка, сервера для тестирования и развёртывания приложений, оргтехника, оборудование и д.р.

Лицензии на программное обеспечение представляют собой специфический вид ресурсов, необходимый для использования тех или иных технологий.

Финансовые ресурсы определяют план и объёмы расходования финансовых средств по этапам выполнения проекта. Основная часть финансирования проекта приходится на заработную плату участникам проекта и расходуется ежемесячно в течение всего времени выполнения проекта. Закупка материальных ресурсов обычно выполняется одновременно в начале соответствующей стадии выполнения проекта. Бюджет проекта будет описывать все статьи расходов и доходов по проекту с указанием периодов их наступления.

7. Сроки проекта. В данном разделе необходимо указать ожидаемые сроки реализации проекта. Как было сказано в предыдущем пункте, точная оценка сроков реализации возможна только после разработки детальной структуры работ проекта. На этапе разработки концепции проекта даётся приблизительная оценка сроков реализации, которая будет уточняться на этапе планирования проекта. Для получения такой оценки можно воспользоваться формулой, выведенной американским инженером и учёным Бари Боэммом: оптимальное время реализации проекта прямо пропорционально кубическому корню от трудоёмкости проекта.

$$T = 2,5 * \sqrt[3]{N}$$

где N – трудоёмкость проекта в человеко-месяцах.

Из данного утверждения есть два следствия:

- Кривая стоимости проекта медленно растёт, если запланированный график проекта длиннее оптимального. При этом работы по проекту всегда занимают всё отведённое для них время.

- Кривая стоимости проекта резко растёт, если запланированный график проекта короче его оптимального графика. При этом ни один проект не может быть завершён быстрее, чем за $\frac{3}{4}$ расчётного оптимального срока реализации не зависимо от количества привлечённых к его реализации ресурсов.

Данные следствия иллюстрируются на рисунке 17.

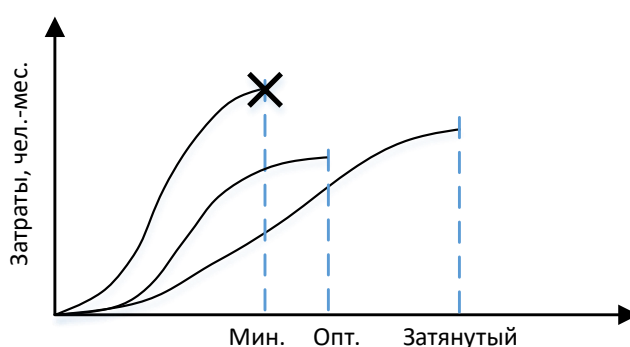


Рис. 17. Следствия закона Бари Боэмма

Кроме определения общего срока проекта в данном разделе должны быть перечислены контрольные точки. Контрольные точки – это важный момент или событие в ходе выполнения проекта, который соответствует завершению определённого этапа работ. Контрольные точки необходимы для оценки хода выполнения проекта и принятия своевременных корректировок для достижения итоговой цели проекта. Каждая контрольная точка должна характеризоваться датой её наступления и критериями, определяющими достижение целевых показателей очередного этапа реализации проекта. В качестве контрольных точек в IT проектах может выступать завершение очередного этапа процесса разработки программного обеспечения или выпуск очередной версии продукта. Продолжительность контрольных точек зависит от типа проекта и объёма выполняемых на этапе работ. Контрольные точки не должны быть слишком частыми. Оптимальная продолжительность этапов определяется от 2 до 8 недель.

8. Риски проекта. Риск – это неопределённое событие, которое в случае его наступления окажет позитивное или негативное воздействие на проект [7]. На этапе разработки концепции проекта в условиях отсутствия детальной информации проводится только качественный анализ рисков и ставится относительная оценка общего уровня рисков. Общий уровень рисков определяется по следующей шкале:

- 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.

Для проектов разработки программного обеспечения уровень рисков на начальном этапе обычно оценивается как выше среднего или высокий. Это связано с тем, что проекты связаны с созданием нового продукта, характеристики которого до конца не известны, а также с применением новых технологий и инструментов.

9. Критерии приёмки. В данном разделе должны быть перечислены основные характеристики системы, которые должны быть продемонстрированы по результатам приёмо-сдаточных испытаний. Данные характеристики должны быть измеримыми и отражать достижение основной цели проекта. Также в данном

разделе может приводиться набор тестов, которые нужно выполнить с продуктом для демонстрации реализации основных функциональных требований к продукту.

10. Обоснование полезности продукта. В данном разделе концепции указываются ожидаемые эффекты от реализации проекта для ключевых заинтересованных сторон:

- Для кого предназначены результаты проекта;
- Описание проблем заказчика, на решение которых направлен данный проект («AS-IS»);
- Описание того, как результаты проекта позволят решить выявленные проблемы («TO-BE»);
- Описание того на сколько существенно решение выявленных проблем для заказчика;
- Описание выгод, которые получит заказчик и исполнитель от реализации данного проекта.

Приведённая структура концепции проекта не является жёсткой и может дополняться другими разделами в зависимости от специфики проекта. Концепция проекта выступает базовым документом в котором определены его основные характеристики и в соответствии с которым будет осуществляться принятие любых решений по проекту на последующих этапах планирования и реализации проекта.

2.5 Концепция проекта разработки программного обеспечения

В данном разделе рассмотрим пример концепции проекта по созданию сервиса поиска схожей по звучанию музыки.

Концепция проекта.

1. Название проекта - “The Same”.
- 2 Цели и результаты проекта.
 - 2.1 Основные цели проекта.
 - 2.1.1 Освоение новых технологий в рамках компании.
 - 2.2 Дополнительные цели.
 - 2.2.1 Освоение рынка стриминга музыкальных произведений.
 - 2.2.2 Повышение репутации компании.
 - 2.2.3 Привлечение большего количества клиентов.

- 2.3 Результаты проекта.
 - 2.3.1 Увеличение прибыли компании.
 - 2.3.2 Кроссплатформенное приложение для стриминга музыкальных произведений и поиска музыкальных произведений по степени их схожести.
 - 2.3.3 ИНС, определяющая схожесть звучания музыкальных произведений.
- 3 Допущения и ограничения.
 - 3.1 Допущения.
 - 3.1.1 Стоимость аппаратного обеспечения не изменится на момент осуществления покупки.
 - 3.1.2 Проектирование реализуемой системы выполняется с помощью UML 2.4.1.
 - 3.2 Ограничения.
 - 3.2.1 Максимальный процент ошибок работы алгоритма - 5%.
 - 3.2.2 Максимальное допустимое количество работающих одновременно пользователей на одном сервере - 100 000.
 - 3.2.3 Используемая в проекте СУБД - MySQL 8.
 - 3.2.4 Фреймворк, используемый для реализации ИНС - TensorFlow (версия фреймворка - 1.11, используемый ЯП - Python 3.6).
- 4 Ключевые участники и заинтересованные стороны.
 - 4.1 Заказчик, исполняющей организацией является “gnp solutions”.
 - 4.2 Спонсор - “gnp solutions”, существующие клиенты.
 - 4.3 Заинтересованные лица.
 - 4.3.1 Музыкальные лейблы.
 - 4.3.2 Спонсоры.
 - 4.4 Пользователи проекта.
 - 4.4.1 Пользователи конкурентных проектов.
 - 4.4.2 Пользователи сети Интернет.
 - 4.5 Куратор проекта – генеральный директор И. И. Иванов.
 - 4.6 Руководитель проекта – ведущий разработчик Ф. Ф. Федеров.
 - 4.7 Соисполнители.
 - 4.7.1 Поставщики аппаратного обеспечения ООО «Computer hardware universe».
- 5 Ресурсы проекта.

- 5.1 Людские ресурсы.
 - 5.1.1 Количество руководителей проекта – 1.
 - 5.1.2 Количество технических лидеров (архитектура, проектирование) – 2.
 - 5.1.3 Количество программистов, занимающихся реализацией ИНС 5 — программисты (реализация ИНС).
 - 5.1.4 Количество программистов, занимающихся разработкой back end части приложения 4.
 - 5.1.5 Количество программистов, занимающихся реализацией мобильных приложений для платформ iOS и Android – 2.
 - 5.1.6 Количество программистов, занимающихся разработкой front end части приложения – 2.
 - 5.1.7 Количество дизайнеров пользовательского интерфейса – 3.
 - 5.1.8 Количество тестировщиков – 3.
 - 5.1.9 Количество экспертов в области ИНС, проводящих обучение имеющегося персонала 1.
- 5.2 Требования к квалификации персонала.
 - 5.2.1 Программисты, занимающиеся front end разработкой должны знать современные фреймворка, практики в CSS3.
 - 5.2.2 Программисты, занимающиеся back end разработкой должны знать такие языки программирования, как Python, Ruby, Cio. Необходимы четкие знания ООП, архитектурных подходов, используемых в приложении.
 - 5.2.3 Программисты, занимающиеся реализацией приложений для мобильных платформ должны владеть основными технологиями и принципами проектирования приложений.
- 5.3 Материальные и другие ресурсы.
 - 5.3.1 Количество серверов для системы контроля версий – 1.
 - 5.3.2 Количество серверов для тестирования – 3.
 - 5.3.3 Количество серверов для обучения ИНС – 1.
- 5.4 Лицензии на ПО.
 - 5.4.1 Проект полностью строится на open source решениях.
- 5.5 Бюджет.
 - 5.5.1 Оплата труда.
 - 5.5.1.1 16000 человеко-часов * 400 Р/час - 6,4 млн Р
 - 5.5.2 Покупка оборудования для серверов тестирования и системы контроля версий.

- 5.5.2.1 4 сервера * 45 000 Р + 180 000 Р.
- 5.5.3 Покупка оборудования для сервера обучения ИНС.
- 5.5.3.1 1 сервер * 600 000 Р - 600 000 Р.
- 5.5.4 Итого: 7,18 млн Р.
- 6 Сроки реализации проекта.
- 6.1 Начало 01.11.2018.
- 6.2 Завершение 01.09.2019.
- 6.3 Контрольные точки:
- 6.3.1 06.11.2018 – утверждение ТЗ.
- 6.3.2 01.12.2018 – завершение обучения персонала.
- 6.3.3 14.12.2018 – завершение разработки дизайна пользователей.
- 6.3.4 14.02.2019 – завершение разработки ИНС.
- 6.3.5 01.03.2019 – начало реализации back end части приложения.
- 6.3.6 01.04.2019 – завершения реализации back end части приложения.
- 6.3.7 02.04.2019 – начало реализации front end части приложения, начало реализации мобильных приложений.
- 6.3.8 14.05.2019 – завершение реализации front end части приложения, начало тестирования и исправления багов в back и front end частях приложения.
- 6.3.9 30.05.2019 – завершение реализации мобильных приложений, начало тестирования и исправления багов в мобильных приложениях.
- 6.3.10 01.07.2019 – начало альфа тестирования приложения.
- 6.3.11 15.07.2019 – завершение альфа тестирования, начало исправления выявленных ошибок.
- 6.3.12 01.08.2019 – начало бета тестирования приложения, исправление ошибок.
- 6.3.13 15.07.2019 – завершение альфа тестирования, начало исправления выявленных ошибок.
- 6.3.14 01.09.2019 – начало эксплуатации системы в промышленных условиях.
- 7 Риски.
- 7.1 В проекте используется новая для организации технология. Из-за отсутствия опыта разработки с использованием данной технологии могут возникнуть проблемы со сроками реализацией. Риски стоит оценить как средние.

- 8 Критерии приёмки
- 8.1 Количество ошибок, совершаемых ИНС не больше 5%.
- 8.2 Время поиска схожих музыкальных произведений не более 15 секунд.
- 8.3 Показатель доступности системы - 98%.
- 9 Обоснование полезности проекта.
- 9.1 Для заказчика.
- 9.1.1 Привлечение новых клиентов.
- 9.1.2 Освоение рынка стриминга музыкальных произведений.
- 9.2 Для компании исполнителя.
- 9.2.1 Освоение нового стека технологий.

Контрольные вопросы.

1. Перечислите основные виды работ, выполняемых на фазе инициации проекта.
2. Какие цели преследует создание модели видения продукта.
3. Какую информацию включает в себя позиционирование продукта?
4. Кто является ключевыми участниками проекта разработки программного обеспечения.
5. Для чего предназначена концепция проекта.
6. Что является целью и задачами проекта, Какие требования к ним предъявляются?
7. Перечислите основные допущения и ограничения, которые применяются к проектам разработки программного обеспечения.
8. Как оценивается трудоёмкость и сроки реализации проекта на этапе построения концепции проекта.
9. Что такое контрольные точки проекта? Какие контрольные точки могут выделяться в IT проектах.
10. Что такое риски проекта и как они оцениваются на этапе инициации проекта?
11. Приведите примеры обоснования полезности IT-проекта для потенциальных заказчиков.

3 Оценка трудоёмкости и экономической эффективности проекта

3.1 Оценка экономической эффективности проекта

Как говорилось ранее, задачей любого проекта является достижение его целей при условии соблюдения имеющихся ограничений. Для оценки того, насколько хорошо проект решает свои задачи, используется понятие эффективности проекта. Эффективность проекта показывает соотношение затрат и результатов реализации проекта применительно к интересам участвующих в проекте сторон. Таким образом, эффективность проекта определяется соотношением затрат времени, финансовых ресурсов и объёма выполненных работ. Данное соотношение подчиняется правилу железного треугольника, представленного на рисунке 18.



Рис. 18. «Железный треугольник» ограничений проекта

В соответствии с правилом железного треугольника ни один из его углов не может быть изменён без влияния на другие компоненты проекта. Следствием из этого правила является тот факт, что есть только один оптимальный вариант распределения ресурсов, который гарантирует наивысшую эффективность проекта.

Учитывая тот факт, что цели проектов различны, для сравнения таких проектов используется общее понятие экономической эффективности, позволяющее оценить проекты разной направленности с финансовой точки зрения. Это связано с тем, что конечной целью деятельности любой компании является

получение финансовой прибыли и любые процессы в жизни человека могут быть измерены в финансовом отношении.

При проведении экономического анализа проекта различают следующие виды показателей эффективности:

1. Показатели коммерческой эффективности. Оценивают экономические последствия от реализации проекта для его непосредственных участников.

2. Показатели бюджетной эффективности. Показывают влияние проекта на государственный, региональный и местный бюджеты.

3. Показатели экономической эффективности. Показывают затраты и результаты проекта, выходящие за пределы прямых финансовых интересов участников проекта.

С точки зрения оценки экономической эффективности проекта наибольший интерес представляет именно первая группа показателей.

При оценке экономической эффективности проекта необходимо учитывать тот факт, что проект реализуется в течении длительного срока, а расходы и доходы по проекту получаются в разные промежутки времени. При этом денежные ресурсы имеют тенденцию к обесцениванию в результате чего одна и та же сумма денежных ресурсов потраченная в разные моменты времени будет иметь различную эквивалентную стоимость. Это связано с процессами инфляции, протекающими в экономике.

Для оценки и сравнения разновременных денежных ресурсов в экономике применяется процедура дисконтирования. В соответствии с этой процедурой будущая ценность сегодняшних денег определяется по формуле:

$$БС = C (1 + d)^t$$

где C – сегодняшняя сумма инвестируемых денежных ресурсов,

$БС$ – эквивалентная сумма денежных ресурсов через период времени t .

d – постоянная норма дисконта, равная приемлемой для инвестора норме дохода на капитал;

t – продолжительность периода инвестирования.

Норма дисконтирования для инвестиций не должна быть ниже нормы инфляции, устанавливаемой в стране, т.к. иначе денежные ресурсы будут обесцениваться. В качестве примера определим стоимость 100000 рублей через три года после их инвестирования

при ставке дисконта 10%. Результаты расчёта представлены в таблице 1.

Таблица 1. Расчёт будущей стоимости денежных ресурсов.

Период инвестирования	Сумма инвестиций, руб.	Коэффициент дисконтирования	Будущая ценность сегодняшних денег, руб.
-	100000	$1,1^0=1$	$100000 * 1 = 100000$
1	100000	$1,1^1=1,1$	$100000 * 1,1 = 110000$
2	100000	$1,1^2=1,21$	$100000 * 1,2 = 121000$
3	100000	$1,1^3=1,33$	$100000 * 1,3 = 133000$

Иногда при экономическом анализе необходимо сравнить не будущую ценность инвестиций, а сумму расходов, приведённую к дате начала проекта. Для этого будет использоваться формула определения сегодняшней ценности будущих денег, которая имеет следующий вид:

$$C = BC / (1 + d)^t$$

В качестве примера рассмотрим сегодняшнюю стоимость 100000 рублей, инвестированных на три года. Результат расчёта приведён в таблице 2.

Таблица 2. Оценка текущей стоимости инвестированных денег.

Срок возврата денег	Долг, руб	Коэффициент дисконтирования	Сегодняшняя стоимость возвращенного долга, руб.
Сегодня	100000	$1,1^0=1$	$100000 / 1 = 100000$
1	100000	$1,1^1=1,1$	$100000 / 1,1 = 90909$
2	100000	$1,1^2=1,21$	$100000 / 1,21 = 82644$
3	100000	$1,1^3=1,33$	$100000/1,33 = 75182$

Далее рассмотрим основные показатели экономической эффективности проектов:

- чистый дисконтированный доход (ЧДД) и интегральный эффект;
- индекс доходности (ИД);
- внутренняя норма доходности (внд);
- срок окупаемости (СО);

Чистый дисконтированный доход определяется как сумма текущих эффектов от реализации проекта, приведённая к

начальному этапу его реализации, как превышение суммарной прибыли от реализации проекта над затратами на его реализацию.

$$ЧДД = \sum_{t=0}^T (R_t - Z_t) \frac{1}{(1+d)^t}$$

Чистый дисконтированный доход рассчитывается по следующей формуле:

где R_t – результаты достигаемые на t -ом шаге расчета;

Z_t – затраты, осуществляемые на том же шаге;

T – горизонт расчета (продолжительность расчетного периода) – равен номеру шага расчета на котором производится закрытие (ликвидация) проекта;

$\Delta = (R_t - Z_t)$ – эффект достигаемый на t -ом шаге расчета;

d – постоянная норма дисконта, равная приемлемой для инвестора норме дохода на капитал.

Под результатами проекта в данном случае понимается прибыль, получаемая на t -ом шаге проекта, а также рыночная стоимость разработанных на этом шаге компонентов программного продукта. Стоимость разработанных компонентов программного продукта можно определить как процент от рыночной стоимости готового продукта после завершения его разработки.

Если чистый дисконтированный доход является положительным, то проект считается эффективным и чем значение выше, тем прибыльность проекта выше. Если чистый дисконтированный доход меньше или равен 0, то проект не эффективен и инвестирование в него приведёт к убыткам.

Для расчёта чистого дисконтированного дохода в проектах с привлечением внешних инвестиций можно воспользоваться альтернативной формулой:

$$ЧДД = \sum_{t=0}^T \frac{R_t - Z_t}{(1+d)^t} - K$$

где K – сумма дисконтированных капиталовложений.

При анализе проектов следует учитывать, что не все проекты являются прибыльными и их чистый доход должен быть положительным. Например, проекты внутренней автоматизации по результатам своей реализации не являются прибыльными. Однако эффект от их внедрения позволит повысить прибыльность других проектов, реализуемых в компании.

Чистый дисконтированный доход позволяет определить чистую прибыль от реализации проекта в абсолютной величине. Однако, данная величина не может использоваться для сравнительного анализа нескольких проектов, т.к. зависит от масштаба проекта. При этом данный показатель не показывает эффективности расходования инвестируемых средств. Для этого используется индекс доходности.

Индекс доходности представляет собой отношение суммы приведённых эффектов к величине капиталовложений. Индекс доходности рассчитывается по следующей формуле:

$$ИД = \frac{1}{K} \sum_{t=0}^T \frac{Rt - Zt}{(1 + d)^t}$$

Индекс доходности является относительной величиной и не зависит от масштаба проекта. Если индекс доходности больше 1, то проект эффективен. Если индекс меньше или равен 1, то проект не эффективен. При этом, чем выше значение индекса доходности, тем выше эффективность инвестирования.

Индекс доходности тесно связан с чистым дисконтируемым доходом. Если ЧДД положительный, то и индекс доходности больше 1. Данный показатель удобно использовать при выборе одного проекта из ряда альтернативных проектов.

Внутренняя норма доходности представляет собой норму дисконта, при которой величина приведённых эффектов равна приведённым капиталовложениям.

$$ВНД = d_1 + \frac{f(d_1)}{f(d_1) - f(d_2)} \times (d_2 - d_1),$$

где d_1 – значение дисконта, при котором ЧДД > 0;

d_2 – значение дисконта, при котором ЧДД < 0;

$f(d_1)$ – значение ЧДД при d_1 ;

$f(d_2)$ – значение ЧДД при d_2 .

Внутренняя норма доходности определяет максимальную величину дисконта, при которой прибыль от проекта превысит расходы на него. Данная величина в дальнейшем сравнивается с требуемой инвестором нормой дохода на вкладываемый капитал.

Срок окупаемости проекта – это минимальный интервал времени, за который доходы от реализации проекта превысят расходы на него. Срок окупаемости может быть рассчитан по формуле:

$$CO = \frac{K}{\bar{D}}$$

где K – суммарный объём затрат на реализацию проекта;

D – ежегодный доход по проекту.

Срок окупаемости показывает, как скоро вложенные в проект инвестиции начнут приносить инвестору прибыль.

Дополнительным показателем, характеризующим эффективность проекта, является коэффициент эффективности инвестиций. Данный коэффициент определяется как отношение чистой прибыли от проекта за всё время его использования к капитальным затратам на его реализацию и поддержку. Формула для расчёта коэффициента эффективности инвестиций имеет следующий вид:

$$КЭИ = \frac{ЧП}{K - ЛС},$$

где $ЛС$ – ликвидационная стоимость проекта

Чем получившаяся величина выше, тем эффективнее инвестиции. Коэффициент эффективности инвестиций лучше рассчитывать с применением процедуры дисконтирования:

$$КЭИ = \frac{\sum_{t=1}^T ЧП_t \times \frac{1}{(1+d)^t}}{\sum_{t=1}^T K_t \frac{1}{(1+d)^t}}$$

Приведённые выше показатели позволяют оценить эффективность реализации проекта и выбрать приоритетные проекты для реализации. При выборе проекта для реализации зачастую возникает проблема сравнения проектов, имеющих разные сроки реализации, что зачастую затрудняет применение приведённых формул расчёта. Алгоритм расчёта в таком случае имеет следующий вид:

1. Найти наименьшее общее кратное от сроков реализации сравниваемых проектов.

2. Рассматривать реализацию проектов в течение найденного срока, при этом считая проекты повторяющимися и реализуемыми несколько раз. При этом показатели проектов будут суммироваться с учётом дисконтирования.

3. Выбрать из проектов тот, для которого суммарная эффективность на выбранном периоде времени будет максимальной.

3.2 Общие принципы оценки трудоёмкости проекта

В литературе даётся следующее определение понятия трудоёмкости проекта – это количество рабочего времени человека, затрачиваемого на производство единицы продукции. Трудоёмкость измеряется в человеко-часах, которые надо затратить на реализацию проекта.

Трудоёмкость является одной из важнейших характеристик проекта т.к. именно она в конечном счёте определяет объём затрат и ресурсов, необходимых для реализации проекта. Оценка трудоёмкости проекта это всегда вероятностная величина, т.к. нельзя точно сказать, сколько времени займёт реализация той или иной работы в рамках проекта. Поэтому вместо расчёта трудоёмкости проекта зачастую используют оценку вероятности того, что трудоёмкость не превысит некоторой фиксированной величины. Распределение вероятности для трудоёмкости проекта представлено на рисунке 19.

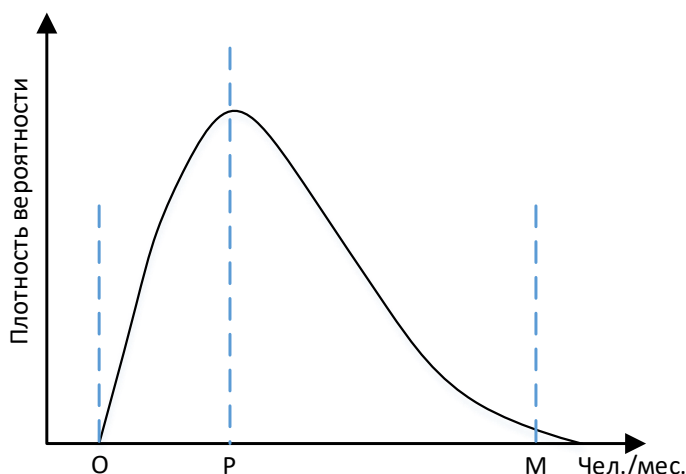


Рис. 19. Распределение вероятности трудоёмкости проекта

Из графика видно, что если P – это наиболее вероятная величина трудоёмкости проекта, то это не значит, что это точная оценка, т.к. вероятность того, что фактическая трудоёмкость превысит данную оценку, составляет более 50%. Как говорилось ранее на начальном этапе анализа проекта хорошей считается такая оценка трудоёмкости проекта, которая находится в пределах от - 50% до +100%.

В условиях недостаточности информации о проекте для получения начальной оценки трудоёмкости проекта зачастую

прибегают к методу экспертной оценки, согласно которому специалисты исходя из своего опыта реализации аналогичных проектов дают приблизительную оценку времени, которое на их взгляд потребуется для реализации данного проекта. В проектах разработки программного обеспечения в качестве таких экспертов выступают непосредственные разработчики, которые в дальнейшем и будут исполнять данный проект. При использовании такого подхода следует учитывать, что в большинстве случаев разработчики склонны занижать оценки времени, необходимые им для выполнения той или иной работы. Это связано с тем, что разработчики до конца не понимают весь объём работ, который предстоит реализовать по проекту, упускают из внимания некоторые детали и сопутствующие работы, которые также требуют затрат времени. В связи с этим в практике управления проектами разработки программного обеспечения есть негласное правило умножать оценку, сделанную разработчиками на 2. В некоторых случаях при реалистичном подходе к оценке даже рекомендуется умножать оценку на число $n=3,14$. В любом случае точность экспертной оценки зависит от квалификации и опыта эксперта, а также уровня знаний и проработанности самого проекта.

При оценке трудоёмкости и сроков реализации проекта необходимо помнить, что чрезмерно оптимистичный прогноз по срокам реализации ведёт к повышению нагрузки на участников проекта и может негативно сказаться на качестве и результатах проекта. Работа в условиях постоянной нехватки времени, оказывает существенное демативирующее влияние на участников проекта, что приводит к высокой текучести кадров, а также низкому качеству выполняемых работ на этапах проектирования и реализации. Это в свою очередь может привести к большому числу ошибок и проблем, выявленных на этапе тестирования и сдачи проекта заказчику. Решение этих проблем на конечной стадии проекта потребует существенных затрат и привлечения дополнительных ресурсов. В результате итоговые затраты по проекту существенно превысят ожидаемые, а иногда могут привести к провалу проекта.

В тоже время существенно завышенные сроки реализации проекта также скажутся негативно на его результативности. В соответствии с законом Паркинсона работа всегда занимает всё

время, отпущенное на неё. Соответственно проект, который мог бы быть выполнен быстрее и с меньшими затратами будет выполняться всегда всё отведённое на него время, что приведёт к необоснованным лишним издержкам.

В связи с этим задача определения трудоёмкости проекта является важной и актуальной.

3.3 Оценка трудоёмкости проекта по методике PERT

Метод PERT (Program Evaluation Review Technique) является одним из наиболее простых и точных методов для оценки трудоёмкости. Данный метод был разработан в 1958 году в ходе проекта по созданию баллистических ракет морского базирования «Поларис». Для применения данного метода необходимо знать точный перечень работ, выполняемых в данном проекте. В связи с этим применение методики бывает затруднительным на начальных этапах планирования проекта.

Для оценки трудоёмкости проекта по методике PERT нет необходимости знать точную оценку трудоёмкости каждой отдельной работы. Достаточно задать оценку трудоёмкости тремя значениями:

M_i – наиболее вероятная оценка трудозатрат;

O_i – минимальные трудозатраты, необходимые на реализацию работы. Это минимальные затраты на реализацию работы при условии что ни один негативный риск не реализовался. Быстрее этой оценки выполнить работу нельзя. Вероятность что реальные трудозатраты будут равны минимальным стремится к нулю.

P_i – максимальная оценка трудозатрат в случае когда все возможные риски реализовались.

Средняя оценка трудоёмкости работы в таком случае будет рассчитываться по следующей формуле:

$$E_i = \frac{P_i + 4M_i + O_i}{6}$$

Среднеквадратичное отклонение определяется по формуле:

$$CKO_i = \frac{P_i - O_i}{6}$$

Если оценки трудоёмкости элементарных работ статистически независимы, а не испорчены, например, необоснованным

оптимизмом то, согласно центральной предельной теореме теории вероятностей суммарная трудоемкость проекта может быть рассчитана по формуле:

$$E = \sum E_i$$

а суммарная оценка среднеквадратичного отклонения составит:

$$CKO = \sqrt{\sum CKO_i^2}$$

Для определения суммарной трудоёмкости проекта, которая не будет превышена с вероятностью 95%, необходимо воспользоваться следующей формулой:

$$E_{95\%} = E + 2CKO$$

Данная оценка является достаточно точной и может быть использована для дальнейшего планирования управления проектом.

Рассмотрим пример оценки трудоёмкости проекта сервиса поиска схожей по звучанию музыки по методике PERT. Для этого составим перечень основных работ, которые должны быть выполнены по данному проекту и оценим их сроки по трём критериям. Результат оценки представлен в таблице 3.

Таблица 3. Оценка трудоёмкости работ по проекту.

Работы	Трудозатраты, чел.-час.		
	минимальные	наиболее вероятные	максимальные
Подготовка ТЗ	48	90	150
Обучение персонала	140	192	236
Закупка необходимого ПО	48	144	200
Проектирование ПО	384	432	500
Разработка ПО	960	1152	1536
Тестирование ПО	96	144	240
Ввод в эксплуатацию	96	130	200
Исправление ошибок	48	192	256
Передача системы в промышленную эксплуатацию	24	48	144

На основе полученных оценок рассчитаем ожидаемые сроки выполнения работ и среднеквадратические отклонения. Результат расчёта представлен в таблице 4.

Таблица 4. Оценка трудоёмкости работ по проекту.

Работы	Ожидаемая продолжительность	Среднеквадратическое отклонение
Подготовка ТЗ	93	17
Обучение персонала	190,7	16
Закупка необходимого ПО	137,3	25,3
Проектирование ПО	435,3	19,3
Разработка ПО	1184	96
Тестирование ПО	152	24
Ввод в эксплуатацию	136	17,3
Исправление ошибок	178,7	34,7
Передача системы в промышленную эксплуатацию	60	20

Суммарная оценка трудоёмкости проекта составит: $E = 2567$ чел.-час. Среднеквадратическое отклонение составит $СКО = 115,13$.

Суммарная трудоёмкость проекта с вероятностью 95% составит $E_{95\%} = 2798$ чел.-часов, что составляет 14,5 чел.-мес.

Согласно закону Барри Боэмма оптимальный срок реализации данного проекта составит: $T = 2,5 * \sqrt[3]{14,5} \approx 6$ месяцев. Оптимальный размер команды разработчиков составит 3 человека.

3.4 Метод функциональных точек для оценки трудоёмкости проекта

Если информация о конкретных работах, которые должны быть выполнены по проекту отсутствует или не полная, то можно применить формальные методы для оценки трудоёмкости. Одним

из таких методов является метод функциональных точек. Анализ функциональных точек - стандартный метод измерения размера программного продукта с точки зрения пользователей системы. Данный метод разработан Аланом Альбрехтом и впервые опубликован в 1979 году. В 1986 году была сформирована Международная Ассоциация Пользователей Функциональных Точек (International Function Point User Group – IFPUG), которая опубликовала несколько версий метода.

Суть метода функциональных точек заключается в оценке сложности и объёма разрабатываемого программного обеспечения с точки зрения количества функционала, востребованного пользователем и поставляемого в конкретной версии программного продукта. Особенность данного метода в том, что он не даёт конкретной оценки трудоёмкости в человекочасах. Вместо этого получается оценка в абстрактных функциональных точках, которые позволяют сравнить сложность и объём проекта с аналогичными проектами, реализованными ранее. Исходя из этого сравнения можно оценить приблизительную трудоёмкость проекта.

Для оценки трудоёмкости проекта по методу функциональных точек нужно выполнить следующие шаги:

1. Определение типа оценки.
2. Определение области оценки и границ продукта.
3. Подсчет функциональных точек, связанных с данными.
4. Подсчет функциональных точек, связанных с транзакциями.
5. Определение суммарного количества не выровненных функциональных точек (UFP).
6. Определение значения фактора выравнивания (FAV).
7. Расчет количества выровненных функциональных точек (AFP).

Процесс оценки представлен на рисунке 20.

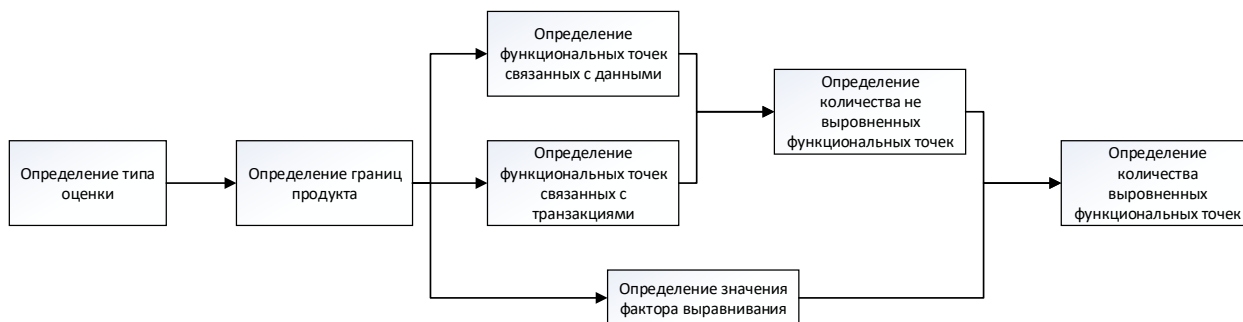


Рис. 20. Процедура анализа трудоёмкости проекта по методу функциональных точек

1. Определение типа оценки. Тип оценки зависит от типа проекта, для которого происходит оценка трудоёмкости. Различают следующие типы оценки:

- Проект разработки – в данном случае оценивается количество функциональных точек, связанных с созданием нового проекта и реализацией всей его функциональности.

- Проект развития – оценка для проект выпуска новой версии уже существующего продукта. При этом отдельно оцениваются функциональные точки, связанные с добавлением, изменением и удалением функционала.

- Продукт – оценивается объём уже разработанного программного продукта.

2. Определение области оценки и границ разработанного продукта. В зависимости от типа оценки различают следующие области оценки:

- Все разрабатываемые функции (для проекта разработки)
- Все добавляемые, изменяемые и удаляемые функции (для проектов поддержки)
- Только функции, реально используемые, или все функции (при оценке продукта и/или продуктов).

Далее определяются границы оценки. Границы позволяют разделить внешние и внутренние данные в системе, а также определить транзакции передаваемые или принимаемые системой. К данным системы относятся:

- Внутренние логические файлы (ILFs) – выделяемые пользователем логически связанные группы данных или блоки управляющей информации, которые поддерживаются внутри продукта.

– Внешние интерфейсные файлы (EIFs) - выделяемые пользователем логически связанные группы данных или блоки управляющей информации, на которые ссылается продукт, но которые поддерживаются вне продукта.

3. Подсчет функциональных точек, связанных с данными. Для подсчёта функциональных точек связанных с данными оценивается сложность данных по следующим показателям:

– DET (data element type) – количество неповторяемых уникальных полей данных. К таким полям относится каждое элементарное поле для ввода информации или элементарное поле структуры данных. Например, форма авторизации содержит два поля – логин и пароль и соответственно количество неповторяемых уникальных полей данных равно 2 DET.

– RET (record element type) – количество логических групп данных. Под логической группой данных понимаются связанные в единую структуру группы элементарных данных. Например поле ФИО это 1 RET, который состоит из 3-ёх полей – фамилия, имя и отчество, т.е. 3 DET.

Сложность данных на основе приведённых показателей оценивается в соответствии с приведённой в таблице 5 матрицей сложности.

Таблица 5. Матрица сложности данных

	1-19 DET	20-50 DET	50+ DET
1 RET	Низкая	Низкая	Средняя
2-5 RET	Низкая	Средняя	Высокая
6+ RET	Средняя	Высокая	Высокая

После оценки сложности данных она пересчитывается в не выровненные функциональные точки в соответствии с нормами пересчёта, представленными в таблице 6. При этом нужно заметить, что количество не выровненных функциональных точек зависит от типа оцениваемого файла – внутренний логический файл (ILF) или внешний интерфейсный файл (EIF).

Таблица 6. Нормы количества функциональных точек в зависимости от сложности данных для внутренних логических файлов (ILF) и внешних интерфейсных файлов (EIF)

Сложность данных	Количество UFP (ILF)	Количество UFP (EIF)
Низкая	7	5
Средняя	10	7
Высокая	15	10

4. Подсчёт функциональных точек, связанных с транзакциями. Транзакция – это элементарный неделимый замкнутый процесс, представляющий значение для пользователя и переводящий продукт из одного состояния в другое. В методе функциональных точек отдельно оцениваются следующие типы транзакций:

- EI (external inputs) – внешние входные транзакции. Это операции по обработке внешних входных данных поступающих в систему из вне.

- EO (external outputs) – внешние выходные транзакции. Это операции, связанные выводом данных и результатов работы программы пользователю. Эти транзакции обычно связаны с обработкой и выводом данных из одного или нескольких внутренних логических файлов ILF.

- EQ (external inquiries) – внешние запросы. Это элементарные операции, которые в ответ на внешний запрос немедленно извлекают и возвращают данные из внутренних и внешних файлов.

Оценка сложности каждого типа транзакций базируется на расчёте следующих характеристик:

- FTR (file type referenced) – количество различных файлов внутренних и внешних, считываемых или модифицируемых в транзакции.

- DET (data element type) – количество элементарных полей данных, затрагиваемых транзакцией. Для внешних входных транзакций это могут быть поля ввода данных, кнопки и т.п., для внешних выходных транзакций это могут быть поля отчёта и т.п., для внешних запросов – поле ввода для поиска данных, поле вывода результатов поиска и т.п.

Сложность транзакций в зависимости от их типа в соответствии с рассчитанными характеристиками рассчитывается на основе матрицы сложности, представленной в таблицах 7, 8.

Таблица 7. Матрица сложности для внешних входных транзакций.

EI	1-4 DET	5-15 DET	16+ DET
0-1 FTR	Низкая	Низкая	Средняя
2 FTR	Низкая	Средняя	Высокая
3+ FTR	Средняя	Высокая	Высокая

Таблица 8. Матрица сложности для внешних выходных транзакций (EO) и внешних запросов (EQ).

EO & EQ	1-5 DET	6-19 DET	20+ DET
0-1 FTR	Низкая	Низкая	Средняя
2-3 FTR	Низкая	Средняя	Высокая
4+ FTR	Средняя	Высокая	Высокая

Количество не выровненных функциональных точек в зависимости от сложности транзакций рассчитывается на основе нормативов, приведённых в таблице 9.

Таблица 9. Нормы количества функциональных точек в зависимости от сложности и типа транзакций

Сложность транзакций	Количество UFP (EI & EQ)	Количество UFP (EO)
Низкая	3	4
Средняя	4	5
Высокая	6	7

5. Определение суммарного количества не выровненных функциональных точек (UFP). Общее количество не выровненных функциональных точек определяется путём суммирования не выровненных функциональных точек связанных с внутренними логическими файлами и внешними интерфейсными файлами, а также функциональных точек, связанных с внешними входными и выходными транзакциями и внешними запросами:

$$UFP = \sum_{ILF} UFP_i + \sum_{EIF} UFP_i + \sum_{EI} UFP_i + \sum_{EO} UFP_i + \sum_{EQ} UFP_i$$

6. Определение фактора выравнивания. Фактор выравнивания определяет влияние не функциональных требований на сложность реализуемого продукта. Значение фактора выравнивания определяется значениями 14 параметров сложности проекта:

- Обмен данными (0 – продукт представляет собой автономное приложение; 5 – продукт обменивается данными по более, чем одному телекоммуникационному протоколу).

- Распределенная обработка данных (0 – продукт не перемещает данные; 5 – распределенная обработка данных выполняется несколькими компонентами системы).

- Производительность (0 – пользовательские требования по производительности не установлены; 5 – время отклика сильно ограничено критично для всех бизнес-операций, для удовлетворения требованиям необходимы специальные проектные решения и инструменты анализа).

- Ограничения по аппаратным ресурсам (0 – нет ограничений; 5 – продукт целиком должен функционировать на определенном процессоре и не может быть распределен).

- Транзакционная нагрузка (0 – транзакций не много, без пиков; 5 – число транзакций велико и неравномерно, требуются специальные решения и инструменты).

- Интенсивность взаимодействия с пользователем (0 – все транзакции обрабатываются в пакетном режиме; 5 – более 30% транзакций – интерактивные).

- Эргономика (эффективность работы конечных пользователей) (0 – нет специальных требований; 5 – требования по эффективности очень жесткие).

- Интенсивность изменения данных (ILF) пользователями (0 – не требуются; 5 – изменения интенсивные, жесткие требования по восстановлению).

- Сложность обработки (0 – обработка минимальна; 5 – требования безопасности, логическая и математическая сложность, многопоточность).

- Повторное использование (0 – не требуется; 5 – продукт разрабатывается как стандартный многоразовый компонент).

- Удобство инсталляции (0 – нет требований; 5 – установка и обновление ПО производится автоматически).
- Удобство администрирования (0 – не требуется; 5 – система автоматически самовосстанавливается).
- Портруемость (0 – продукт имеет только 1 инсталляцию на единственном процессоре; 5 – система является распределенной и предполагает установку на различные «железо» и ОС).
- Гибкость (0 – не требуется; 5 – гибкая система запросов и построение произвольных отчетов, модель данных изменяется пользователем в интерактивном режиме).

Приведённые параметры DI (degree of influence) оцениваются по относительной шкале в диапазоне от 0 до 5. Далее рассчитывается суммарный эффект TDI (total degree of influence) путём суммирования рассчитанных факторов:

$$TDI = \sum DI$$

Фактор выравнивания VAF рассчитывается по следующей формуле:

$$VAF = (TDI * 0,01) + 0,65$$

7. Расчёт количества выровненных функциональных точек. Количество выровненных функциональных точек зависит от типа проекта и значения фактора выравнивания. Для оценки уже разработанного проекта используется следующая формула:

$$AFP = UFP * VAF$$

Данная оценка учитывает только непосредственную функциональность, реализованную в проекте.

Для оценки проекта разработки нового программного обеспечения используется показатель DFP (development functional point), который учитывает кроме функциональных точек, связанных с непосредственной функциональностью продукта (UFP) ещё функциональные точки для дополнительной функциональности, связанной с установкой и настройкой программного продукта (CFP – conversion functional points). Получается следующая формула для расчёта:

$$DFP = (UFP + CFP) * VAF$$

Проект доработки существующего программного продукта или выпуска новой версии оценивается показателем EFP (enhancement functional point). В данной случае в расчёте учитываются функциональные точки, связанные с добавляемой

функциональностью (ADD), функциональные точки, связанные с изменённой функциональностью (CHGA), функциональные точки, связанные с дополнительной функциональностью (CFP) и функциональные точки, связанные с удалённой функциональностью (DEL). Кроме того в данном случае рассчитывается два фактора выравнивания: VAFA – величина фактора выравнивания, рассчитанного после внесения изменений в проект и VAFB – величина фактора выравнивания, рассчитанного до начала выполнения проекта. Итоговая формула для расчёта количества выровненных функциональных точек имеет следующий вид:

$$EFP = (ADD + CHGA + CFF) * VAFA + (DEL * VAFB)$$

Суммарное влияние фактора выравнивания лежит в диапазоне $\pm 35\%$ от рассчитанного количества не выровненных функциональных точек.

Расчёты по приведённому алгоритму метода функциональных точек ничего не говорят о реальной трудоёмкости проекта. Для оценки трудоёмкости проекта в человеко-часах необходимо определить среднюю трудоёмкость реализации одной функциональной точки и умножить на рассчитанный показатель.

В качестве примера рассмотрим оценку в функциональных точках формы поиска схожего по звучанию трека, которая представлена на рисунке 21

Поиск треков

Критерии поиска

Жанр
Options 1 DET

Альбом
1 DET

Исполнитель
1 DET

Продолжительность
1 DET

Трек образец

File path 1 DET

Название трека

Найти 1 FTR

Результаты



Название трека 1 DET

Исполнитель 1 DET

Альбом 1 DET

Жанр 1 DET

1 DET

1 DET

1 DET

1 DET

▶ ● 🔊 🔊

Рис. 21. Форма поиска треков.

Рассчитаем сложность данных для представленной формы. Данные описываются двумя составными записями: Критерии поиска, состоящие из четырёх элементарных полей и трек образцов, состоящий из одного поля. Таким образом, данные состоят из 2 RET и 5 DET. В соответствии с таблицей оценка сложности данных для приведённой формы будет Low. Данной оценке соответствует 5 не выровненных функциональных точек. Для оценки сложности транзакций можно сделать вывод что на форме представлена одна транзакция, что соответствует 1FTR. Данная транзакция извлекает из внутренних файлов 4 поля – 4 DET. Сложность данной транзакции также будет оценена как Low и будет соответствовать 3 не выровненным функциональным точкам. Суммарная оценка приведённого интерфейса в не выровненных функциональных точках составит $5 + 3 = 8$ не выровненных функциональных точек. Если принять значение каждого параметра характеристик продукта за значение 3, то фактор выравнивания для оцениваемого продукта составит $FAF = (0.01 * \sum_{i=1}^{14} 3) + 0,65 = 1,07$. Тогда количество выровненных функциональных точек будет равно $8 * 1,07 \approx 9$.

3.5 Оценка трудоёмкости проекта разработки программного обеспечения на основе методики COSOMO II

Методика COSOMO была впервые опубликована в 1989 году американским инженером и учёным Барри Боэмом. В её основе лежит модель оценки трудоёмкости, полученная при анализе результатов 63 проектов компании «TRW Aerospace». В 1997 году данная методика была усовершенствована и получила название COSOMO II. Параметры модели были рассчитаны на основе анализа 161 проекта в сфере разработки программного обеспечения.

Особенностью предложенной методики является её относительная простота по сравнению с аналогичными методиками расчёта трудоёмкости. Для проведения расчёта трудоёмкости необходимо знать только приблизительную оценку объёма разрабатываемого программного кода в тысячах строк. В основе модели лежит формула регрессии с параметрами, основанными на особенностях анализируемого проекта и стандартных

коэффициентах. Простота данного подхода компенсируется сравнительно низкой точностью оценки, т.к. слабо учитывается специфика конкретного проекта и применяются типовые коэффициенты, полученные на основе анализа статистики. Данный подход может применяться на начальных стадиях планирования проекта, когда нужна грубая приблизительная оценка трудоёмкости.

В методике COCOMO II различается две стадии оценки проекта:

1. Предварительная оценка на начальной стадии планирования проекта;
2. Детальная оценка – на этапе проработки детального плана реализации проекта.

Расчёт трудоёмкости по методике COCOMO II производится по следующим формулам:

$$PM = A \times SIZE^E \times \prod_{i=1}^n EM_i ,$$

$$A = 2,94 ,$$

$$E = B + 0,01 \times \sum_{j=1}^5 SF_j ,$$

$$B = 0,91$$

где: PM – трудоёмкость проекта, чел./мес.;

SIZE – размер продукта в тысячах строк кода, KSLOC;

EM_i – множители трудоёмкости проекта;

SF_j – Факторы масштаба проекта.

Основным параметром, используемым при расчёте трудоёмкости проекта по методике COCOMO II является размер продукта. Размер продукта оценивается в тысячах строк кода и имеет величину KSLOC (Kilo Source Lines Of Code). Он может быть оценён экспертным путём, на основе сравнения проекта с аналогичными или рассчитан с применением метода PERT. При расчёте размера продукта необходимо учитывать, что он зависит от языка программирования, на котором продукт будет реализовываться. В таблице 10 представлены статистические оценки количества строк кода, необходимых для реализации одной функциональной точки на разных языках программирования.

Таблица 10. Оценка количества строк кода, необходимых для реализации одной функциональной точки на разных языках программирования.

Язык программирования	Оценка количества строк		
	Наиболее вероятная	Оптимистичная	Пессимистичная
Assembler	172	86	320
C	148	9	704
C++	60	29	178
C#	59	51	66
Java	61	50	100
JavaScript	56	44	65
VisualBasic	50	14	276

После расчёта размера программы, необходимо оценить факторы масштаба проекта, которые определяют сложность проекта, наличие опыта и слаженность команды разработки. Факторы масштаба оцениваются по относительной шкале от уровня Very Low, до уровня Extra High. Всего в методике COCOMO II используется 5 факторов масштаба:

1. PREC – прецедентность, наличие опыта аналогичных разработок. Very Low – опыт в продукте и платформе отсутствует; Extra High – продукт и платформа полностью знакомы;

2. FLEX – гибкость процесса разработки. Very Low – процесс строго детерминирован; Extra High – определены только общие цели;

3. RESL – архитектура и разрешение рисков. Very Low – риски неизвестны / не проанализированы; Extra High – риски разрешены на 100%;

4. TEAM – слаботанность команды. Very Low – формальные взаимодействия; Extra High – полное доверие, взаимозаменяемость и взаимопомощь;

5. PMAT – зрелость процессов. Very Low – CMM Level 1; Extra High – CMM Level 5.

В зависимости от выбранной оценки каждого фактора масштаба выбирается его числовое значение на основе коэффициентов, приведённых в таблице 11.

Таблица 11. Значения факторов масштаба проекта

Фактор масштаба	Оценка уровня фактора					
	Very Low	Low	Nominal	High	Very High	Extra High
PREC	6,20	4,96	3,72	2,48	1,24	0,00
FLEX	5,07	4,05	3,04	2,03	1,01	0,00
RESL	7,07	5,65	4,24	2,83	1,41	0,00
TEAM	5,48	4,38	3,29	2,19	1,10	0,00
PMAT	7,80	6,24	4,68	3,12	1,56	0,00

После оценки факторов масштаба, необходимо оценить множители трудоёмкости проекта. В зависимости от стадии оценки проекта используется разное количество множителей трудоёмкости. На стадии предварительной оценки применяется 7 множителей трудоёмкости, на стадии детальной оценки – 17. Множители трудоёмкости оцениваются по той же относительной шкале, что и факторы масштаба. На стадии предварительной оценки определяются следующие множители трудоёмкости:

1. PERS – квалификация персонала. Extra Low – аналитики и программисты имеют низшую квалификацию, текучесть больше 45%; Extra High - аналитики и программисты имеют высшую квалификацию, текучесть меньше 4%;

2. RCPX – сложность и надежность продукта. Extra Low – продукт простой, специальных требований по надежности нет, БД маленькая, документация не требуется; Extra High - продукт очень сложный, требования по надежности жесткие, БД сверхбольшая, документация требуется в полном объеме;

3. RUSE – разработка для повторного использования. Low – не требуется; Extra High – требуется переиспользование в других продуктах;

4. PDIF – сложность платформы разработки. Extra Low – специальные ограничения по памяти и быстродействию отсутствуют, платформа стабильна; Extra High – жесткие ограничения по памяти и быстродействию, платформа нестабильна;

5. PREX – опыт персонала. Extra Low – новое приложение, инструменты и платформа; Extra High - приложение, инструменты и платформа хорошо известны;

6. FCIL – оборудование. Extra Low – инструменты простейшие, коммуникации затруднены; Extra High –

интегрированные средства поддержки жизненного цикла, интерактивные мультимедиа коммуникации;

7. SCED – сжатие расписания. Very Low – 75% от номинальной длительности; Very High – 160% от номинальной длительности.

Числовые значения множителей трудоёмкости оцениваются на основе данных, приведённых в таблице 12.

Таблица 12. Значения множителей трудоёмкости проекта

Множитель трудоёмкости	Оценка уровня фактора						
	Extra Low	Very Low	Low	Nominal	High	Very High	Extra High
PERS	2,12	1,62	1,26	1,00	0,83	0,63	0,50
RCPX	0,49	0,60	0,83	1,00	1,33	1,91	2,72
RUSE	n/a	n/a	0,95	1,00	1,07	1,15	1,24
PDIF	n/a	n/a	0,87	1,00	1,29	1,81	2,61
PREX	1,59	1,33	1,22	1,00	0,87	0,74	0,62
FCIL	1,43	1,30	1,10	1,00	0,87	0,73	0,62
SCED	n/a	1,43	1,14	1,00	1,0	1,00	n/a

Из приведённой таблицы, в частности, следует, если в нашем проекте низкая квалификация аналитиков, то его трудоемкость возрастет примерно в 4 раза по сравнению с проектом, в котором участвуют аналитики экстра-класса.

Рассмотрим пример оценки трудоёмкости проекта сервиса поиска схожей по звучанию музыки по методу СОСОМО II. В качестве оценки количества строк кода возьмём оценку 3000 строк. Оценим факторы масштаба и трудоёмкости (таблица 13).

В результате трудоёмкость проекта составит 30 чел.-мес.

Таблица 13. Оценки факторов масштаба и трудоёмкости.

Фактор	Оценка	Значение
PREC	Nominal	3,72
FLEX	High	2,03
RESL	Low	5,65
TEAM	Very High	1,1
PMAT	Nominal	4,68
PERS	Nominal	1,62
RCPX	Very High	0,6
RUSE	Nominal	1
PDIF	Very High	1,91
PREX	Low	1
FCIL	Very High	1,81
SCED	Nominal	1,22

Контрольные вопросы.

1. В чём заключается правило «железного треугольника»?
2. Для чего применяется метод дисконтирования?
3. Какие показатели используются для оценки экономической эффективности проекта?
4. Как производится сравнение проектов, имеющих разные сроки реализации?
5. В каких единицах измеряется трудоёмкость проектов.
6. Расскажите алгоритм оценки трудоёмкости проекта по методике PERT.
7. В чём заключается метод функциональных точек?
8. Расскажите алгоритм оценки трудоёмкости проекта по методике COSOMO II.

Литература

1. Попов, Ю.И. Управление проектами: учеб. пособие / Ю. И. Попов, О.В. Яковенко; Институт экономики и финансов "Синергия". - М.: ИНФРА -М, 2015. - 208 с.
2. Свод знаний по управлению проектами [Электронный ресурс]. – URL: <http://www.pmtoday.ru/project-management/pmbok-pmp/pmbok.html>
3. Бовин, А.А. Управление инновациями в организациях: учебное пособие/ А.А.Бовин, Л.Е. Чередникова, В.А. Якимович. 2-е изд. - М.: Омега -Л, 2011. - 415 с.
4. Товб А. С., Ципес Г. Л.. Управление проектами: стандарты, методы, опыт. - М.: Олимп-Бизнес, 2008.
5. Войку, И.П. Управление проектами: конспект лекций. / И.П. Войку. – Псков: Псковский государственный университет, 2012. – 204 с.
6. Информационные технологии управления. Учебное пособие / Под ред. Ю.М. Черкасова. - М.: ИНФРА-М. 2001. - 218 с. ил.
7. Шкурко В.Е., Управление рисками проектов : [учеб. пособие] / В. Е. Шкурко ; [науч. ред. А. В. Гребенкин] ; М-во образования и науки Рос. Федерации, Урал. федер. ун-т. – Екатеринбург : Изд-во Урал. ун-та, 2014. – 184 с
8. Управление высокотехнологичными программами и проектами / Рассел Д. Арчибальд ; Пер. с англ. Мамонтова Е. В. ; Под ред. Баженова А. Д., Арефьева А. О. – 3-е изд., перераб. и доп. – М. : Компания АйТи ; ДМК Пресс, 2010. – 464 с., ил.
9. Лапыгин Ю. Н. Управление проектами: от планирования до оценки эффективности. — М.: Омега-Л, 2008. — С. 252.
10. Том ДеМарко. Deadline. Роман об управлении проектами. — М: Вершина, 2006. — С. 143.

Учебное издание

Ужаринский Антон Юрьевич

Конюхова Оксана Владимировна

**УПРАВЛЕНИЕ ПРОГРАММНЫМИ ПРОЕКТАМИ В IT КОМПАНИЯХ.
ЭТАПЫ ИНИЦИАЛИЗАЦИИ И ПЛАНИРОВАНИЯ**

Учебное пособие

Печатается в авторской редакции

Технический редактор
Лукиянов П.В.

Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Орловский государственный университет имени И.С. Тургенева»

Подписано к печати _____. Формат 60х90 1/16
Усл.печ.л. _____. Тираж ____ экз.
Заказ № _____

Отпечатано с готового оригинал-макета