[**Лекция 7 Управление программным проектом - введение** 2](#_Toc496549639)

[**7.1 Общие вопросы управления программными продуктами** 2](#_Toc496549640)

[**7.1.1 Некоторые определения** 2](#_Toc496549641)

[**7.1.2 О типичных рисках ИТ-проектов** 5](#_Toc496549642)

[**7.2 Измерения и формализация программного проекта** 7](#_Toc496549643)

[**7.2.1 Два измерения деятельности программного проекта** 7](#_Toc496549644)

[**7.2.2 Понятие формализации управления ИТ-проектом** 7](#_Toc496549645)

[**7.2.3 Этапы формализации программного проекта** 9](#_Toc496549646)

[**7.2.4 Применение формализации в ИТ-проектах** 11](#_Toc496549647)

[**7.3 Итоги** 11](#_Toc496549648)

## **Лекция 7 Управление программным проектом - введение**

### **7.1 Общие вопросы управления программными продуктами**

#### **7.1.1 Некоторые определения**

***Управление*** – процесс воздействия на объект управления (систему, процесс, проект и т.п.), ведущий к достижению поставленной цели управляющим субъектом.

Сопутствующие понятия (факторы управления):

* Наличие цели управления;
* Осуществимость управляющего воздействия;
* Возможность измерений состояния объекта;
* Ограниченность управления! (время, финансы, качество).

***Проект*** – протяжённое во времени предприятие (комплекс действий), направленное на создание уникальных продуктов, услуг или иных результатов, с точной датой начала и окончания действия, отличающееся от продолжающихся, повторных действий и требующее прогрессивного совершенствования характеристик (определение PMI – Project Management Institute).

***Характеристики проекта***

* **Цель проекта** – чётко выраженный конечный результат проекта, определённый в сопутствующих терминах затрат, качества и времени реализации;
* **Уникальность** – проект – это разовое начинание, которое не будет повторяться. Сомнение – а как же здания, которые строятся по одному архитектурному проекту (документации). Документация на здание одна и та же, но для разных зданий могут значительно отличаться среды реализации и используемые ресурсы (ещё пример – студенты одной специальности, но разного года поступления);
* **Ограниченность во времени** – есть начало и конец. Важно для концентрации и высвобождением ресурсов;
* **Ограниченность ресурсов**;
* **Сложность** – необходимость решения множества задач, отношения между которыми могут быть довольно сложными – последовательность, скорость выполнения, ресурсоёмкость;
* **Неопределённость** – результат (достижение цели в назначенные сроки при использовании только выделенных ресурсов) не гарантирован из-за наличия рисков;
* **Предсказуемость** – так как проект реализуется по определённому жизненному циклу, то, несмотря на риски, определённая степень предсказуемости наличествует.

***Категории управления проектами***

Категории управления проектами отражают основные понятия теории и практики этой деятельности. К ним относятся:

* Цели;
* Критерии успеха и ограничения: стоимость, сроки, качество;
* Основные рычаги управления: ресурсы и технологии;
* Вспомогательные рычаги управления: контракты, организация, взаимодействие, персонал;
* Неопределённость, связанная с рисками выполнения проекта;
* Треугольник ограничений проекта.

Ключевая категория, «участвующая» в процессе управления проектами – ограничения.

Правило Лермана и «железный» треугольник управления проектом.

**Закон технологии Лермана его следствия**

Любую техническую проблему можно решить, имея достаточно времени и денег.

Следствия:

1.Вам всегда не будет хватать либо времени, либо денег.

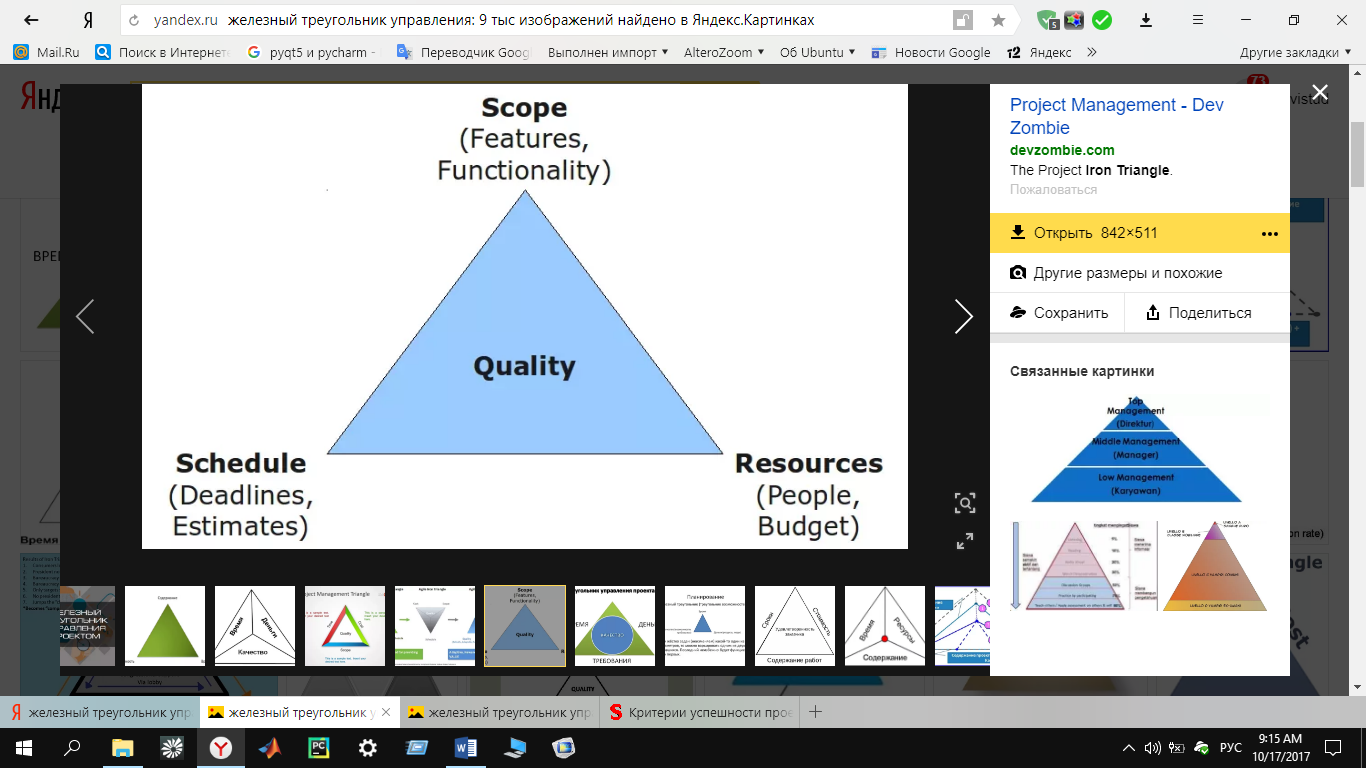
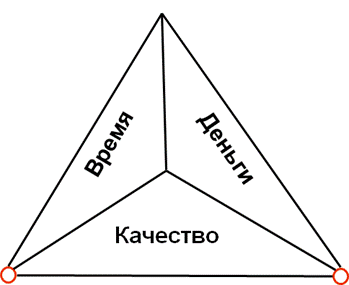
2. Финансирующие организации откажутся от вашего предложения, если его результаты не будут известны заранее.

**(Закон Мэрфи и его следствия и вариации – www. Firststep.ru )**

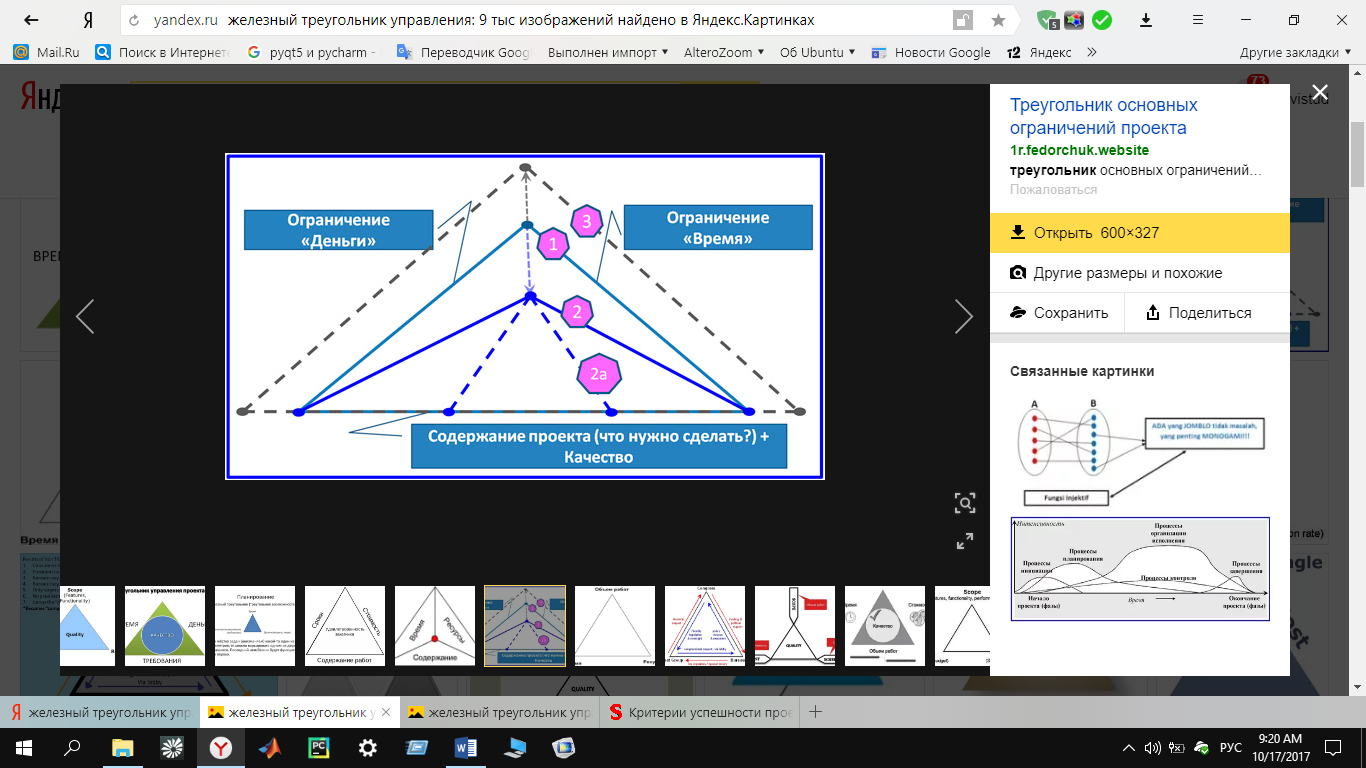
Правило железного треугольника:

Задача проекта — достижение конкретной бизнес-цели, при соблюдении ограничений «железного треугольника» (рис. 7.1). Это означает, что ни один из углов треугольника не может быть изменен без оказания влияния на другие. Например, чтобы уменьшить время, потребуется увеличить стоимость и/или сократить содержание.

Центр и верхняя вершина фиксированы. Нижние могут перемещаться. Площади внутренних треугольников пропорциональны качеству проекта, его бюджету и времени выполнения.



а) б)



с)

Рис 7.1 – Треугольник связи трёх параметров выполнения проекта: время, качество (ресурсы), ресурсы.

а) точка в центре внешнего треугольника – объём работ; б) и с) – качество пропорционально площади треугольника

Наиболее важная причина провала проекта – плохое управление, которое может быть обусловлено следующими причинами:

* Маловразумительными целями (нечёткими, плохо определёнными);
* Плохим планированием (?);
* Необходимостью использования новых малознакомых технологий;
* Отсутствием методологии управления проектом (?);
* Плохим учётом возможных рисков, что выражается в конечном счёте отсутствием резервов (ресурсов) для их преодоления.

#### **7.1.2 О типичных рисках ИТ-проектов**

<https://lektsia.com/1x1402.html>

Перечень актуальных рисков, которыми действительно управляют, как правило, колеблется в пределах 10 – 20 на проект. В этот список попадают наиболее «актуальные» риски.

Можно выделить 5 наиболее распространенных рисков, из-за которых происходит большинство расхождений между планом и реальным ходом проекта:

1. внутренние изъяны календарного планирования;

2. изменение требований;

3. текучесть кадров;

4. нарушение спецификаций;

5. низкая производительность.

Рассмотрим наиболее распространенный риск «Изъяны календарного планирования». Он появляется из-за изъянов процесса планирования бюджета времени и средств. Ошибки календарного планирования состоят в неправильной оценке размеров продукта, который предстоит создать: могут быть упущены какие-либо работы, переоценка размера работ, оказавшаяся в плане, редко оказывается достаточной, чтобы компенсировать недооценку. Оценки календарного планирования основаны всего лишь на требованиях заказчика и чаще всего расходятся с итоговым сроком выполнения проекта. Когда календарное планирование строится без учета размера продукта, весьма вероятен перерасход времени на 50-80%.

Компании, которые прилагают усилия для определения размера продукта, могут сократить влияние ошибок календарного планирования до 15% и менее. Соответственно, заложен необходимый резерв времени, который высчитывается всё более точно с учетом опыта предыдущих проектов.

Ключевой момент в данном риске состоит в том, что когда проект не укладывается в график, то это происходит несмотря на работу разработчиков, а не благодаря ей.

Риск **«Изменение требований**» возникает в связи с тем, что область деятельности заказчика не остается статичной за время создания программного обеспечения. Она будет изменяться со скоростью, диктуемой рынком и собственными темпами технологического развития, а, следовательно, меняются и желания заказчика относительно проекта. С точки зрения проекта, эти изменения всегда являются раздуванием требований. Даже удаление того, что уже создано — своего рода раздувание, поскольку требует дополнительной работы.

На основании различных крупных проектов был сделан вывод о том, что размер **разумно ожидаемых изменений составляет менее 1% в месяц**. Компания-разработчик ПО должна предполагать такую возможность и иметь соответствующие ресурсы для удовлетворения изменений требований.

Следующим риском является «Текучесть кадров».

Оценка текучести кадров основана на следующих данных:

• средний процент текучести технического персонала в компании;

• оценка общих потерь времени при найме для каждой замены;

Общие потери времени - число месяцев, которое потребуется нанятому работнику на достижение того же уровня производительности, который был у замененного им работника. Обычно это время составляет от 2 месяцев на самых простых позициях в IT-отделах до 24 месяцев для компании, производящей очень сложные приложения. Длина этого периода зависит от того, насколько сложна область и насколько она отличается от опыта и навыков, наличие которых можно ожидать от начинающего разработчика.

Четвертый риск «Нарушение спецификаций» дискретный, если он реализуется, то это почти всегда губительно для проекта. Нарушение спецификаций относится к краху процесса переговоров по установлению требований, которые идут в начале проекта.

Распространенной попыткой «решения» проблемы является оттягивание решения, маскировка и утаивание проблемы с помощью двоякой трактовки спецификации.

Хотя возможно описать (специфицировать) продукт двусмысленно, но невозможно создать продукт неоднозначным. В итоге приходится столкнуться с отложенными проблемами, и конфликт разгорается вновь. В худшем случае это происходит на очень поздней стадии проекта, когда потрачены почти все отведенные на проект ресурсы (деньги и время). Проект очень уязвим в этой стадии, и отказ любой из сторон поддерживать его может привести к быстрому прекращению.

Нарушение спецификаций проявляется и в случае, когда недовольные участники проекта перегружают проект все большим и большим количеством функций. Такого рода «заваливание» обычно происходит в конце работы по анализу и приводит к невозможности договоренности по спецификациям.

Проявление этого риска вероятно в любом проекте, пока не происходит четкого прекращения прений по поводу спецификаций. После чего риск можно убрать.

Следующим риском является «**Низкая производительность**».

Команды разработчиков могут отличаться производительностью, но, в целом, скорость работы одинакова, если не подвержена влиянию других рисков. Однако для очень малых команд это неверно. Например, команда из одного человека подвержена куда большему воздействию низкой или высокой производительности.

Сбалансированный риск, вроде низкой или высокой производительности, просто вносит шум в процесс. Он расширяет диапазон неопределенности без сдвига среднего ожидаемого показателя, в каком бы то ни было направлении.

Данные риски можно использовать для оценки того, был ли процесс управления рисками осуществлен разумно. Если пять вышеперечисленных рисков не были учтены, то вряд ли можно говорить об управлении рисками.

Далее: стратегические риски, финансовые риски, риски опасностей.

### **7.2 Измерения и формализация программного проекта**

#### **7.2.1 Два измерения деятельности программного проекта**

Первое измерение – в области **инженерии**. Связано с проектированием, программированием, тестированием и т.п.

Второе измерение – области **управления**. Связано с надлежащим планированием и контролем действий инженерии. Именно с этими действиями связано успешное выполнение проекта (с должным качеством, в запланированными стоимостью и сроком выполнения).

#### **7.2.2 Понятие формализации управления ИТ-проектом**

7.2.2.1 Определение формализма и формализации

(фр. formalisme, от лат. formalis — относящийся к форме).

1) Предпочтение, отдаваемое форме перед содержание в различных сферах чел. деятельности.

В области человеческих отношений **Формализм** проявляется в неукоснительном следовании правилам этикета, обряда, ритуала, даже в тех случаях, когда жизненная ситуация не требует этого, делает его бессмысленным, комичным или драматичным.

2) Одно из основных направлений в основаниях математики и логики, объединившее идею строгой формализации математических рассуждений с абстрактным подходом к математике.

3) В социально-гуманитарных науках — методологический принцип отказа от содержательного анализа изучаемых объектов в пользу формально-логического анализа.

Таким образом, можно встретить многим непонятный термин «формализация», причем в самых разных областях науки и техники. Тем, кто хочет расширить горизонты своих знаний, желательно понять, что такое формализация.

Что такое формализация с **научной** точки зрения в **общем понимании**?

Затронем немного научный аспект. Будем отталкиваться от того, что слово формализация происходит от слова "формальность", то есть является условным, а иногда даже абстрактным понятием, **позволяющим объяснить природу несуществующего объекта или явления и спрогнозировать его свойства в определенной среде при заданных начальных условиях**.

Лингвистика любого современного языка абсолютно не совпадает с выражением логических форм или природой мышления. Таким образом, логика сама по себе вынуждена использовать некие отвлеченные понятия, чтобы описать то или иное явление. Так и появляется относительное понятие формальности происходящего.

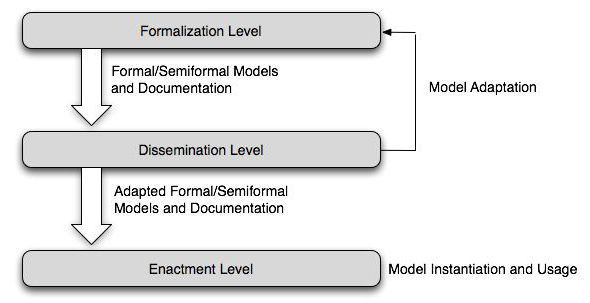


Рис. 7.2 Диаграмма последовательности создания и внедрения проекта

Формальные/неформальные модели → уровень распространения → уровень принятия (конкретизация и применение)

Как уже нетрудно догадаться, суть формализации сводится к тому, чтобы описать или предопределить некие свойства объекта или процесса (даже не существующего на данный момент) и спрогнозировать его применение в случае появления в реальном мире. Но это общее представление. Само понятие формализации куда шире.

Основные методы формализации - это прогнозирование и моделирование. Применяются такие технологии исключительно для получения конечных данных об объектах или процессах, которые не известны, но их можно предположить и с высокой точностью рассчитать.

Если посмотреть на виды формализации, практически все они сводятся только к логическим умозаключениям и вычислениям. Читателю не составит труда провести параллель между компьютерным моделированием, доказательством теорем и т. д. на основе аксиом и постулатов.

#### **7.2.3 Этапы формализации программного проекта**

Если рассматривать компьютерные системы, **то первым этапом формализации является описание процесса.** Но здесь не используются инструменты обычного языка (буквы, слова, словосочетания, предложения). Создать определенную [математическую модель](http://fb.ru/article/50819/matematicheskaya-model-etapyi-proektirovaniya) можно только с использованием некоего алгоритма на основе выбранного языка программирования, но только после постановки общей задачи.

Иными словами, при моделировании поведения объекта или процесса суть происходящего нужно описать чисто математическими символами, применив математический алгоритм.

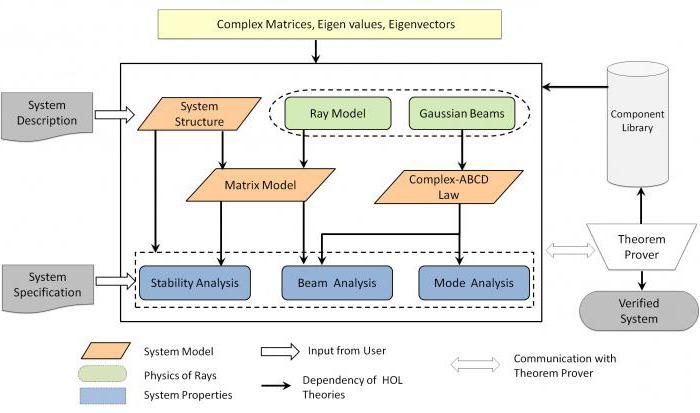
**Результатом формализации** является получение анализа действительного предсказуемого события, которое последует после того, как исследуемая технология будет применена на практике или определенный природный процесс войдет в стадию реального проявления.

Далее следует **концептуализация** поставленной задачи. Здесь есть два варианта:

* в первом случае это определение подхода в виде использования атрибутов и признаков;
* второй вариант подразумевает применение когнитивного анализа, не говоря уже о постановке задачи, сбора начально используемых данных, условий и т. д.

**После определения понятий и начальных условий** изучаются существующие взаимосвязи между объектами и процессами, а также так называемые семантические отношения, подразумевающие использование методики локального представления.

**Далее следует обработка начальных данных на основе выбранного алгоритма**, после чего выдается результат с указанием процента погрешности. Как правило, она не превышает 5%, а в большинстве случаев результат вероятности доходит до 99%. Любой человек или машина все равно оставляют «запас прочности» на [непредвиденные обстоятельства,](http://fb.ru/article/197654/obstoyatelstva-nepredvidennyie-vozniknovenie-nepredvidennyih-obstoyatelstv) ведь абсолютно все учесть невозможно.



**Зачем все это нужно?**

Если разобраться, такие принципы позволяют производить анализ поведения объектов и процессов. Иными словами, можно предугадать, как будет развиваться тот или иной процесс.

Результат математического моделирования во многом (если не на все сто процентов, то с вероятностью до 95%) может стать весомым аргументом в пользу выпуска современной техники, поможет предсказать погоду, даже спрогнозировать общественное поведение как реакцию на события в мире.

**А в области создания ПО?**

#### **7.2.4 Применение формализации в ИТ-проектах**

Простые проекты – высокий уровень формализма не требуется!

Сложные проекты – требуется уровень формализма для обеспечения строгости планирования и отслеживания запланированных задач на основе количественных подходов (метрик).

Формализация требует применять чётко определённые процессы, и результаты зависят от устойчивости этих процессов. Использование при этом количественных подходов усиливает этот подход и делает его более определённым.

Технически использование формализованного подхода при решении задачи заключается в последовательном выполнении определённых шагов (этапов), заранее определённых для принятой методологии. При этом рекомендовано накапливать и использовать опыт, полученный в успешных проектах.

Существует опасность излишней бюрократизации процесса управления, отставания используемой методологии управления от развития технических средств. Поэтому сами методологии непрерывно развиваются, рождая новые подходы и направления реализации

### **7.3 Итоги**

1. Процессы инженерии определяют, как выполнить работу.
2. Процессы управления проектом определяют, как установить контрольные точки, организовать персонал, управлять рисками, отслеживать продвижение вперёд.
3. Менеджеры проектов должны стремиться придерживаться процессного подхода, поскольку процессы подразумевают и представляют по сути коллективное знание.
4. Не прибегая к процессам нельзя предсказать результаты проекта.