

Занятие 4: Лонгрид

Сайт: [Your It Career Hub](#)
Курс: Project Management
Книга: Занятие 4: Лонгрид

Напечатано:: Alex Sidorenko
Дата: суббота, 22 марта 2025, 13:46

Оглавление

1. Введение. О чем этот урок
2. Классический подход к управлению
3. PMbOK
4. Как управляется ИТ-проект
5. Задание 1
6. Решение 1
7. Преимущества PMBoK
8. Задание 2
9. Решение 2
10. Гибкий подход к разработке
11. Задание 3
12. Решение 3
13. Подведем итоги
14. 🖋️ Упражнение
15. Решение для упражнения

1. Введение. О чем этот урок

Мы разобрались с тем, как выстроены отношения в команде и кто за что отвечает. Теперь необходимо выбрать, каким методом управлять разработкой. Этим занимается ИТ-менеджером Ольга 🧐. Её задача - выбрать наиболее эффективный способ управления, чтобы написать посадочную страницу для маркетинговой кампании. Мы вместе с ней пройдем этот путь и выясним, какой метод нам подойдет. Готовы? Давайте посмотрим на план занятия.

В этом уроке мы:

1. Изучим разницу между классическим и гибким подходами;
2. Рассмотрим популярные фреймворки управления в ИТ;
3. Разберемся в методах приоритезации задач;
4. А также решим учебный кейс.

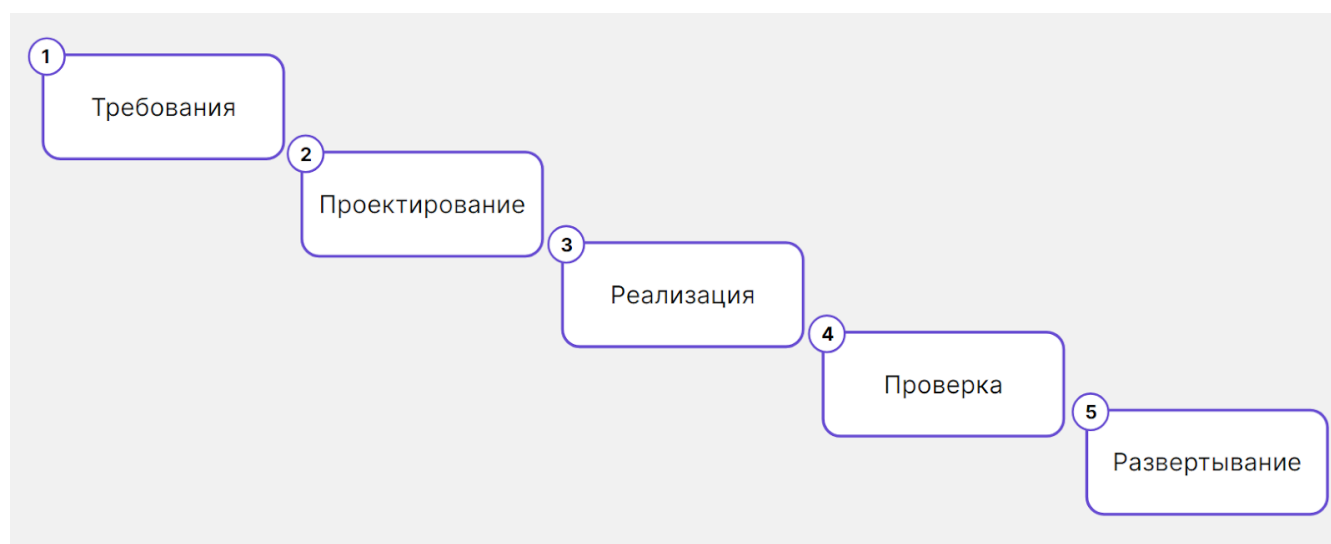
2. Классический подход к управлению

Управление в ИТ появилось в индустрии относительно недавно. Причина простая - возраст отрасли. Если дома люди строят несколько тысяч лет, то код пишут всего 60. Поэтому экспертизу по строительству условных пирамид применили к разработке программного обеспечения. Если очень коротко, то это и есть классический подход. Но давайте все же дадим конкретное определение.

Классический подход к управлению разработкой заключается в том, что вы последовательно реализуете этапы проекта. Например, сбор требований, проектирование, реализация, проверка и развертывание результатов работ Заказчику. Только в конце проекта готовый продукт передается Заказчику.

Одним из таких классических подходов является **фреймворк Водопад**. Давайте разберём его подробнее.

Фреймворк состоит из пяти этапов: требования, проектирование, реализация, проверка и развертывание. Каждый этап фреймворка двигает исполнение проекта вперед. Любое изменение в требованиях к проекту запускает процесс с самого верха. Называется метод Водопадом, потому что требования “падают” сверху вниз, превращаясь в продукт.



1. **Требования.** Самый первый этап, во время которого определяют и анализируют требования проекта: системные требования, требования к ПО, пожелания заказчика и т. д. На основе всей этой информации создают входную документацию, где описывают, что должно получиться в итоге. Короче, на этом этапе пишут первую, обобщенную, версию ТЗ.
2. **Проектирование.** Когда первая версия ТЗ готова и есть общее понимание, что нужно

сделать, команда приступает к проектированию — детализирует ТЗ, согласует с заказчиком логику работы системы, описывает, что и как будет работать. На выходе этого этапа всё ещё не проясняется вопрос реализации, но уже становится примерно понятно, сколько нужно людей и часов на работу.

3. **Реализация.** Затем команда окончательно проясняет, как именно будет происходить разработка проекта, с помощью каких инструментов (языков программирования, оборудования, сервисов и т. д.). Каркас, который проработали на предыдущих этапах, становится более целостным, потихоньку формируется облик продукта. На этот этап приходится большая часть работы над проектом.
4. **Проверка.** На этом этапе проводят полноценное тестирование продукта, чтобы найти и исправить критические (и не очень) проблемы.
5. **Развертывание.** И, наконец, когда всё протестировано и отлажено, переходят к последнему этапу, в рамках которого сдают проект заказчику, устанавливают, внедряют — в общем, вводят продукт в эксплуатацию. Кроме того, сюда может входить и последующее сопровождение, и поддержка, в том числе техническая.

3. PMbOK

Есть специальный документ - **PMbOK (project management book of knowledge)**. Этот документ описывает классические методы управления, разработанные институтом PMI (Project Management Institute). Данный свод знаний описывает лучшие практики по минимизации рисков по доставке ценности, а также обеспечения прозрачности процесса.

Преимущества и недостатки PMBoK

PMBoK наиболее эффективен в проектах двух видов:

1. **Типовые проекты, где все процессы понятны и предсказуемы.** Есть измеримая цель, к которой нужно прийти, и понятен путь достижения этого результата.
Пример такого проекта — строительство жилого дома. Команда знает, какой дом строить, из каких этапов состоит строительство, какие ресурсы понадобятся. Такой проект легко разложить на этапы: проведение геодезических работ, подготовка проекта, строительные работы, подключение коммуникаций, отделка и передача объекта заказчиком, — а потом реализовать их в рамках процессов, предусмотренных PMBoK.
2. **Мегапроекты с тысячами этапов, в которых нужна строгая координация.** Например, строительство олимпийских объектов.

Не все проекты складываются так, что их конечная цель видна в самом начале. Или бывает, что есть цель, но не всегда виден путь, который нужно пройти, чтобы этой цели достичь. Это проекты с высокой степенью неопределённости, PMBoK не будет в них эффективен. К ним можно отнести:

1. **Проекты запуска новых продуктов — стартапы,** где неизвестно, какой результат проекта будет в итоге успешным. Команда стартапа на протяжении всего проекта ищет оптимальный результат, постоянно меняет цели и пути их достижения.
2. **Проекты НИОКР (научно исследовательские).** В таких проектах тоже нет чётких целей и задач. В рамках исследований учёные ищут новые свойства, материалы, новые технологии. Работу таких проектов нельзя описать на этапе подготовки.

Если исходить из этого, **главный минус PMBoK в том, что он не универсален.** Он применим в проектах с понятным результатом и планом достижения цели и не применим там, где нужно искать результат в процессе или нащупывать оптимальный путь к достижению цели.



Этот недостаток PMBoK отчасти нивелирован в последней актуальной версии, которая

включает много знаний из сферы гибких методологий. В целом PMI движется в сторону того, чтобы избавиться от этого минуса полностью.

Преимущества PMBOK

PMBOK предоставляет простую и понятную систему для продвижения по жизненному циклу проекта.

PMBOK создает стандартизированный процесс управления проектами.

PMBOK создает множество данных и документации, которые могут быть использованы для постоянного улучшения текущего проекта и будущих проектов.

Недостатки PMBOK

PMBOK может потребовать от многих членов команды охватить все области знаний.

При чрезмерной стандартизации некоторые члены команды будут чувствовать себя менее мотивированными к творчеству

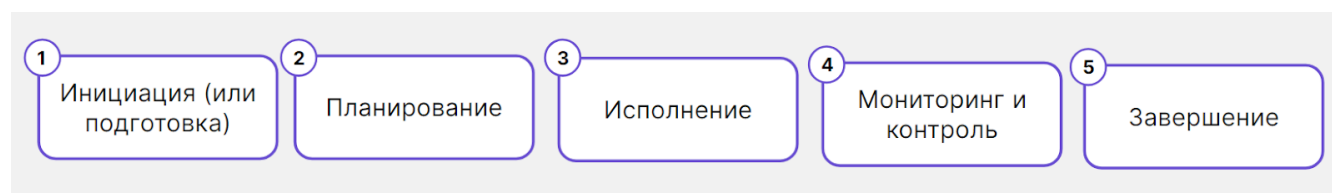
Выполнение всех процессов и шагов может показаться трудоемким

4. Как управляется ИТ-проект

Управление ИТ-проектом по PMbOKI ставит перед командой две задачи:

1. Удовлетворить потребности заказчика.
2. Реализовать проект в срок в рамках установленного бюджета и с требуемым содержанием.

Модель жизненного цикла программного обеспечения предусматривает конкретные шаги:



- **Инициация (или подготовка).** Включает в себя процессы запуска нового проекта. Это постановка целей проекта и проведение предварительных исследований. В результате получается база, необходимая для старта проекта и его дальнейшего развития.
- **Планирование.** Процессы этой группы уточняют цели проекта и детально прорабатывают действия для их достижения. Это постоянные процессы, которые сопровождают весь проект. Если изначально спланировать проект по одному сценарию, а в результате сбоя всё пойдёт не по плану, нужно разрабатывать новый сценарий. Такой сбой может произойти на любом этапе проекта — соответственно, на любом этапе может понадобиться планирование.
- **Исполнение.** Включает в себя процессы выполнения проекта и контроля его ресурсов — людей, технологий, материалов. Например, к этой группе процессов относят набор и развитие команды проекта, проведение закупок, управление реализацией проекта.
- **Мониторинг и контроль.** Процессы этой группы идут параллельно с процессами планирования и исполнения. Они предполагают непрерывную проверку соблюдения планов проекта — сроков, качества, стоимости.
- **Завершение.** Эти процессы формализуют передачу результатов проекта и закрывают его. На этом этапе составляют финальные отчёты, передают результат заказчику, формируют окончательную документацию и распускают команду.

5. Задание 1



Как вы думаете, для каких проектов подходит РМбОК?

6. Решение 1

Ответ: Типовые проекты, где все процессы понятны и предсказуемы. Есть измеримая цель, к которой нужно прийти, и понятен путь достижения этого результата. Пример такого проекта — строительство жилого дома.

7. Преимущества PMBoK

PMBoK помогает сделать такие проекты предсказуемыми и прозрачными, выстроить все процессы в оптимальной последовательности. Так, чтобы возникающие простои не приводили к большим финансовым затратам: например, когда тысячи строителей не могут выполнять свою работу, потому что другой этап проекта не успели сдать в срок.

Вы же помните ИТ-менеджера Ольгу 🧑‍💻? Она является ИТ-менеджером в команде Антона. Именно такой метод она выбрала, чтобы написать типовую посадочную страницу для маркетинговой кампании постоянного Клиента. Все известно заранее, требования не меняются, а команда уже выполняла такое. Поэтому этот способ показал наибольшую эффективность. Рамочный договор на месяц подписан, теперь пора за работу!

8. Задание 2



Как вы думаете, с какого этапа начинается разработка проекта по РМбОК?

9. Решение 2

Ответ: с этапа Инициация.

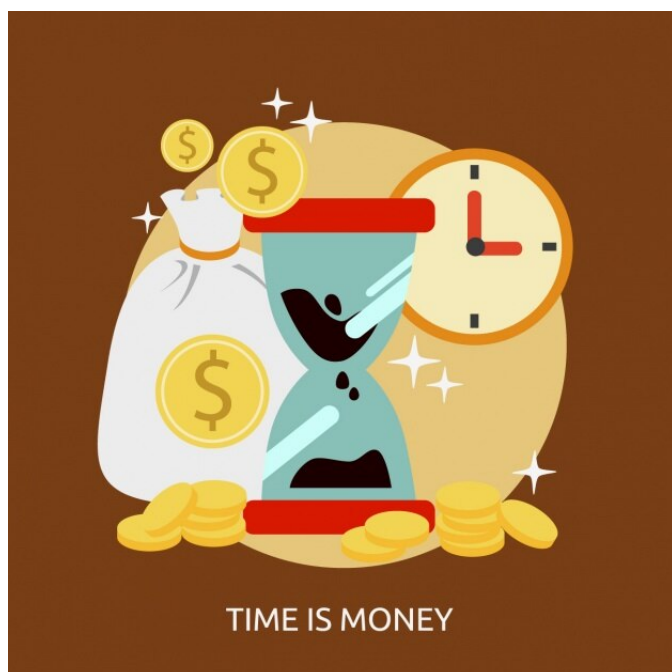
10. Гибкий подход к разработке

Мы только что разобрали классический подход к управлению проектами, у него есть свои недостатки и преимущества. Однако его жесткость и прозрачность гарантирует соблюдение сроков по сложным проектам.

У всего есть антипод, так и тут. Для сложных проектов без понятного результата лучше использовать гибкие методологии.

Гибкая методология — это процесс управления, который основан на разделении проекта на более мелкие задачи и этапы. Такое дробление позволяет командам учитывать отзывы всех заинтересованных сторон, переоценивать результаты работы и применять циклический подход на каждом этапе процесса.

Почему эти методы стали эффективны? Все просто: со временем разработка стала только дороже, а необходимость эффективно двигаться на рынке стала возрастать. Проверка гипотезы гибким методом банально дешевле, чем классическая разработка всего решения целиком.



Ну а теперь вернемся к нашим старым знакомым аналитикам Кириллу и Алене. Они получили расплывчатое требование от Заказчика:

“Нужно срочно разработать модуль рекомендаций для сервиса бронирования билетов.”

Выглядит как сложная бизнес-логика выбора билетов и отображения информации для конечного пользователя. Что же делать?

Срочный созвон с тим-лидом Анастасией и принимается решение: проверить пару гипотез относительно решения данной задачи. Для этого Антону нужно попробовать написать прототип сервиса с использованием стандартных библиотек. В итоге стало ясно, что изначальная оценка была неправильной, а обещанию Заказчику о выполнении такого задания пришлось отзываться.

Ребята бы не попали в такую ситуацию, если бы использовали фреймворк управления SCRUM. Однако это уже тема для следующего урока.

11. Задание 3



Как вы думаете, может ли программист принимать участие в формировании продуктовых гипотез?

12. Решение 3

Ответ: Да, может. Чем меньше компания, тем чаще происходит такое расслоение. Но и в успешных компаниях программисты работают над продуктом с аналитиками.

13. Подведем итоги

На этом уроке вы:

1. Изучили разницу между классическим и гибким подходами;
2. Рассмотрели популярные фреймворки управления в ИТ;
3. Посмотрели разницу в подходах.

На этом всё. Теперь выполните практику, чтобы закрепить материал. А после этого переходите к следующему уроку, где мы поговорим о одной из самых популярных гибких методологий - SCRUM.

14. 🖋️ Упражнение

Ольга с командой должна срочно закрыть небольшой проект: за 3 месяца надо собрать посадочную страницу. Используя этапы классического подхода, опишите порядок разработки проекта.

15. Решение для упражнения

Ответ:

1. Сбор БТ.
2. Оценка работ.
3. Согласование работ.
4. Отрисовка дизайна.
5. Разработка бэк-енда.
6. Разработка фронт-енда.
7. Тестирование.
8. Сдача проекта.