



# Управление производственным процессом разработки программного обеспечения



# Содержание

1. Планирование целей
2. Декомпозиция целей и выделение MVP
3. Определение зависимостей
4. Планирование исполнения, выбор инструментов
5. Оценка длительности работ
6. Оценка срока с учетом ресурсов проекта
7. Организация исполнения и контроль
8. Планирование качества
9. Планирование внедрения
10. Кроме того: работа с рисками, управление изменениями, метрики успешности проекта

# Планирование – инструмент реализации проекта в условиях ограничений

## Планирование –

это процесс определения целей проекта, разработки стратегии их достижения и документирования последовательности действий, необходимых для успешного выполнения проекта (определение из PMBoK)



# Проектирование – предусловие планирования

## Проектирование –

это процесс определения целей, критериев успеха, обоснования необходимости проекта, создания видения результата, шагов его достижения, технической реализации, оценки необходимых ресурсов. Выполняется на этапе инициации проекта

### Шаги создания концепта:

1. Выявляются ключевые заинтересованные лица
2. Анализируются потребности и требования
3. Проводится исследование рынка и конкурентов
4. Формулируются цели и задачи проекта
5. Формируется визуальный образ результата
6. Проводится оценка рентабельности результата
7. Оцениваются ресурсы для выполнения проекта
8. Проводится презентация и утверждение концепции

# Инструменты проектирования. CJM

## Customer Journey Map (CJM) –

это карта взаимодействия пользователя с продуктом, карта составленная с учётом конкретных действий пользователей, с учетом его эмоций, целей, мотивов, привязки его к конкретному этапу, препятствий пользователя. Визуализация его опыта

**Abstract**



# Инструменты проектирования. USM

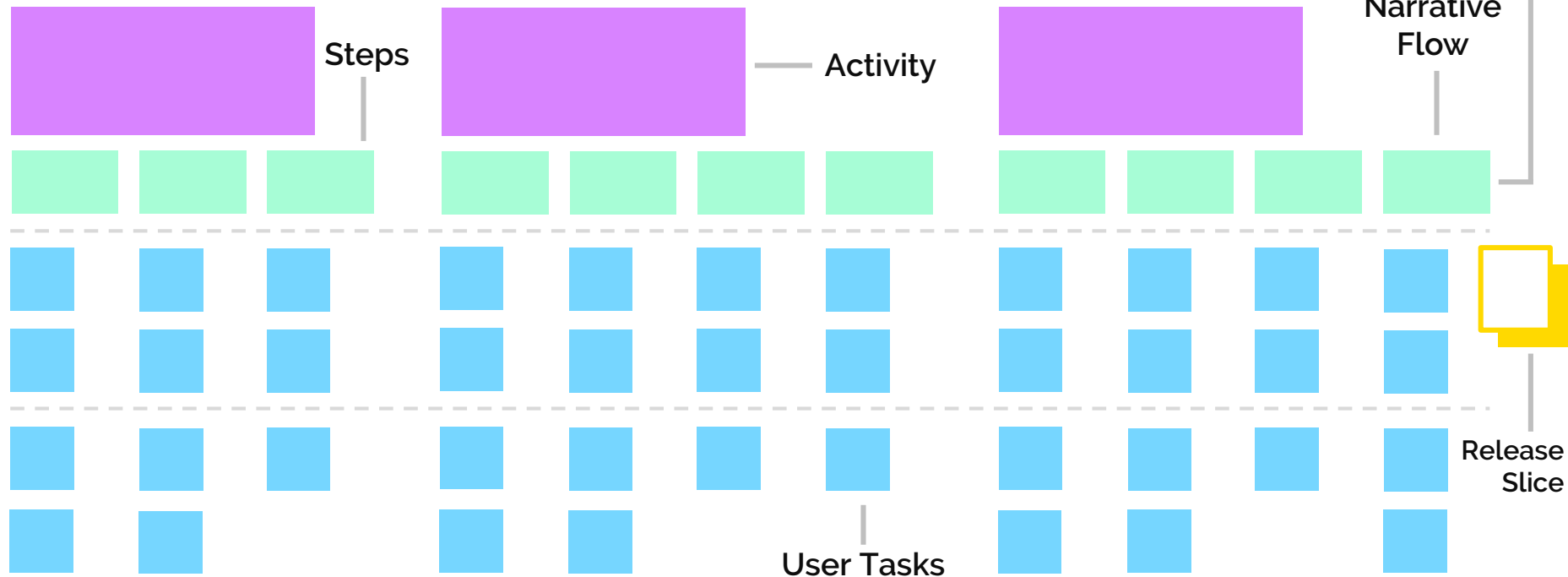
## User Story Mapping (USM) –

инструмент визуализации пользовательских историй. Он показывает, как выглядит взаимодействие пользователей с продуктом: где они чувствуют себя уверенно, а где - нет, какие функции представляют особую ценность, а какие - наоборот, можно смело убирать

# Инструменты проектирования. USM



— User & User goal

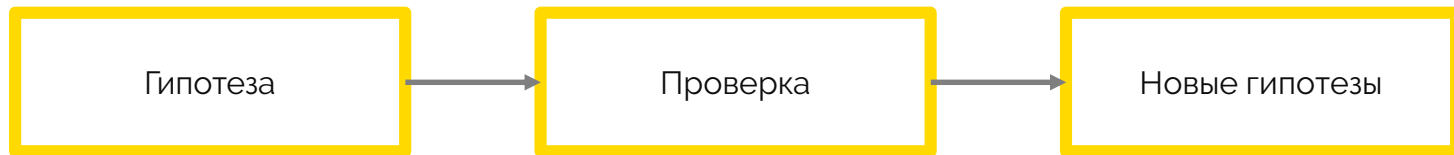




# Инструменты проектирования. PoC



Бывает, что неопределенность  
не удастся разрешить



## PoC (Proof of Concept) –

минимально возможный прототип, иллюстрирующий  
реализуемость задачи в рамках выбранного подхода

# Выявление минимальной версии продукта

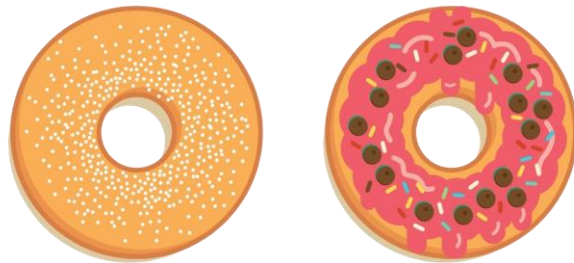
## Для чего нужно:

- Быстро проверить гипотезу
- Получить отзыв пользователей
- Снять или выявить риски на ранней стадии
- Уточнить план, сфокусироваться на главных целях

**Проверить идею на практике  
и более уверенно  
масштабировать продукт**

## MVP (Minimum Viable Product) –

это минимально жизнеспособный продукт, который представляет собой самую раннюю версию продукта с минимальным набором функций, достаточным для презентации публике и проверки на первых потребителях



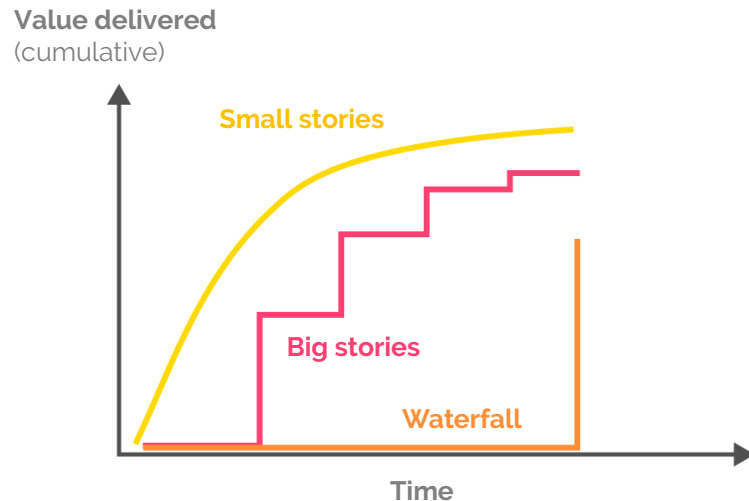
# Декомпозиция – способ поэтапного достижения цели проекта

1. Необходима для составления плана
2. Позволяет оценить проект
3. Учитывает и продуктовые и процессные задачи – любые работы на проекте
4. Минимизирует риски: **ошибки оценки, отсутствия обратной связи, актуальности требований, потери знаний, усложнения**

Мало этапов/задач — страдает точность плана

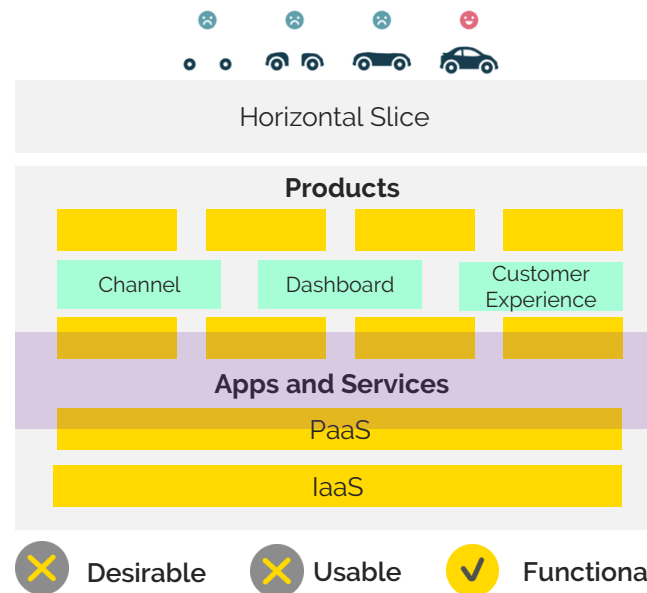
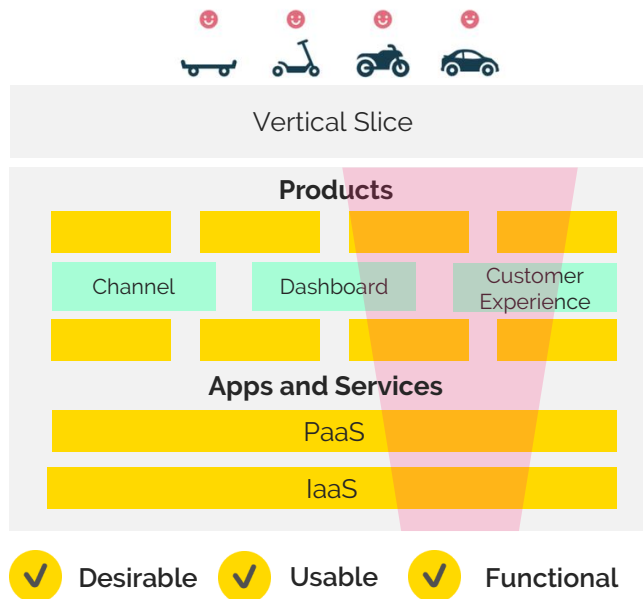
Много этапов/задач — страдает достоверность

Гранулярность может изменяться со временем



# Ключевые подходы к декомпозиции

1. **Функциональная** – по подсистемам и модулям, функциям системы
2. **Сквозная (вертикальная)** – по сценариям пользователя в системе



# Критерии завершенности задач

## Критерии приемки (Acceptance Criteria)

Определяются **владельцем** продукта, применяются **к сценарию**, чтобы определить его готовность с точки зрения получения желаемой **ценности**

### Пример:

- пользователь может войти в систему, используя код из СМС
- кнопка «отправить» становится активна при заполнении всех полей формы

## Критерии готовности (Definition of Done)

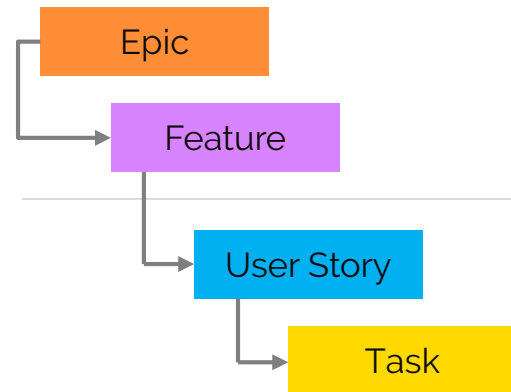
Определяются **командой**, применяются **к задачам**, историям, чтобы проверить **исполнение обязательных этапов** при разработке. Общее правило «готовности» работы

### Пример:

- задача развернута на тестовом окружении
- проведено тестирование по чек-листу
- актуализирована документация

# Уровни декомпозиции

Уровень	Описание	Горизонт
<b>Эпик</b>	Значимая часть функционала. Новый функциональный блок. Это может быть целиком MVP или его половина, треть	Месяцы
<b>Фича</b>	Элемент декомпозиции Эпика. Законченный функциональный сценарий или возможность	1-3 спринта
<b>Стори</b>	Элемент декомпозиции Фичи. Минимальный элемент бизнес-поставки (заметный конечному пользователю)	2 недели
<b>Задача</b>	Элемент декомпозиции Стори. Атомарная работа одному исполнителю. Минимальный элемент технической поставки	1-3 дня



**Важны все уровни декомпозиции – помогают повышать точность и достоверность плана**

# «Скрытые» задачи проектов разработки

При планировании разработки можно легко забыть о задачах, не несущих **видимый прямой продуктовый результат**:

1. Подготовка инфраструктуры разработки
2. Задачи интеграции со смежными командами
3. Задачи управления и коммуникации
4. Подготовка документации
5. Подготовка и миграции данных

**Смежные команды/продукты/системы** – связаны с проектом, без взаимодействия с ними невозможно решение задач и достижение целей проекта

## Пример:

- общий справочник пользователей в организации
- стороннее хранилище данных
- регламенты передачи проекта в эксплуатацию



**Все эти задачи так же включаются в план, оцениваются, получают исполнителей**

# Иерархическая структура работ (ИСР)

## ИСР –

инструмент планирования, позволяющий разделить большой проект на иерархию небольших задач.

Глубина иерархии произвольная

### ✓ Признаки хорошей ИСР:

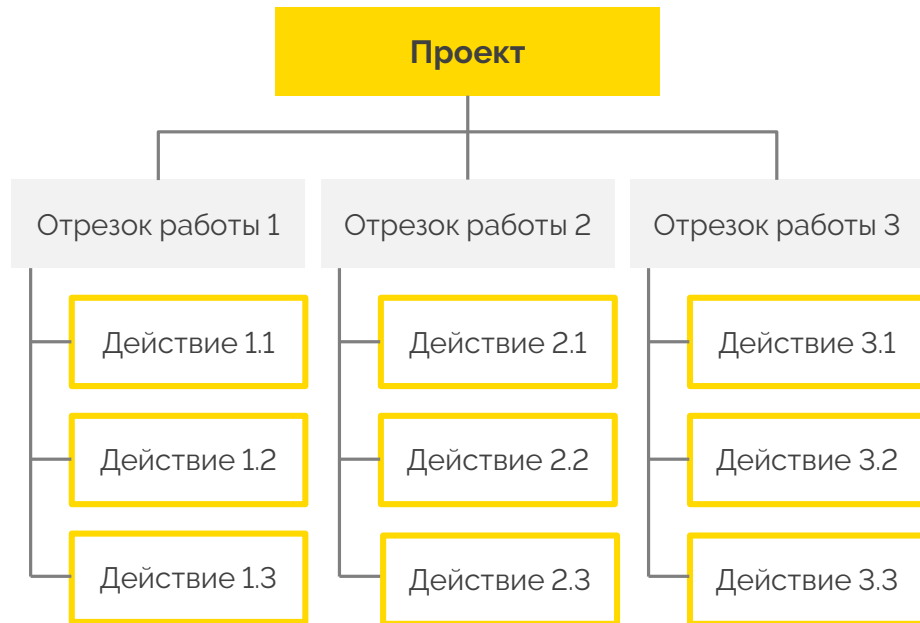
- Содержит все задачи проекта
- Одна задача – один ответственный
- Дерево задач (одна задача – один родитель)
- Блоки равнозначные и автономные
- Схема однозначная
- Все задачи примерно одного размера

✦ Мобильный регистратор
✦ Прототип: новый стиль, без связи с бэкендом
▷ Дизайн
▷ Параметризация экранов уникальным стилем в зависимости от логина
▷ Управление
✦ MVP: продажа багажа
▷ Подготовка
▷ Инфраструктура
✦ Разработка
✦ Android клиент
▷ Авторизация в приложение
▷ Тестирование и отладка продажи услуг с новым бэкендом
▷ Разработка экранов
▷ Оплата
▷ Фискальные операции и передача в ОФД- параметризация данными продавца
▷ Мониторинг
▷ Backend
▷ Админка для управления клиентами и пользователями
v.1: + выбор места
✦ Тестирование решения
Тестирование методов сервера
Тестирование приложения
Тестирование проводки фискальных платежей и смена ФН
✦ Документация
Создание пользовательской документации
✦ Поддержка
▷ Организация поддержки



# Оформление ИСР

- 1. Контурная структура.** Это просто нумерованный список, первый уровень декомпозиции — это 1, второй — 1.1, третий — 1.1.1, и т. д. В списке все пункты расположены друг под другом и идентифицируются по номеру.
- 2. Иерархическая таблица.** Есть уровень декомпозиции, порядковый номер элемента, название, описание, ответственный, сроки, стоимость, зависимости (ссылка на пункт таблицы, с которым у текущего пункта есть связь). Всё это сведено в таблицу и позволяет суммировать работы и результаты в план-график и бюджет по итогу.
- 3. Древовидная структура.** Самая наглядная форма представления ИСР, потому что разветвление показано очень просто от корня (проекта) к ветвям (блокам) и веточкам поменьше (результатам), листикам (задачам). Как вариант — реализовывать в виде майндмэпа, выписывая только результаты



# INVEST – критерии хорошей истории



**I**

**Independent**

независимая



**N**

**Negotiable**

понятная



**V**

**Valuable**

ценная



**E**

**Estimable**

способна  
быть  
оцененной



**S**

**Small**

компактная




**T**

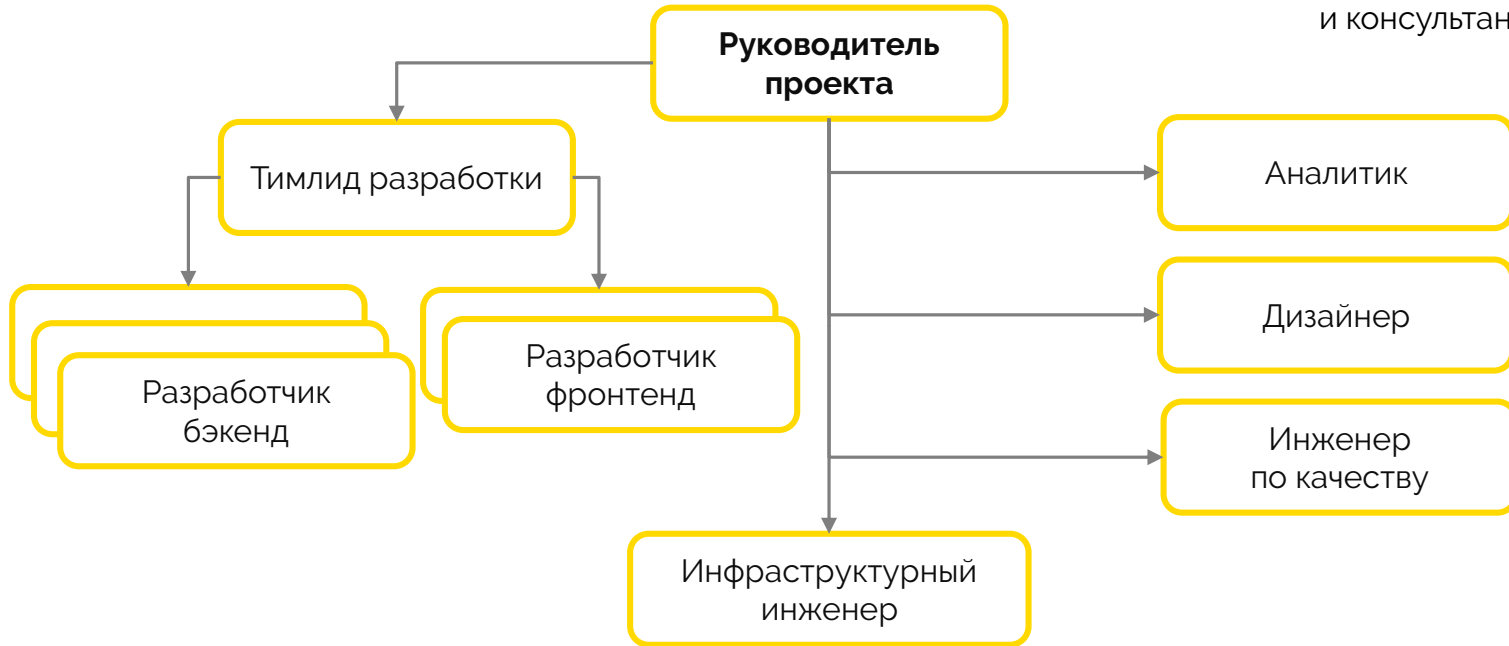
**Testable**

тестируемая

# Состав команды

**Команда проекта разработки** – группа специалистов разного профиля, которые вовлечены в работу над ним

Обычно это временная группа, создаваемая на период выполнения проектных задач, которая может также включать внешних исполнителей и консультантов 



# Оценка должна учитывать исполнителя

## Роли исполнителей в проекте:

1. Разработчики разных видов
  - Backend
  - Frontend
  - iOS
  - Android
  - И т.д.
2. QA инженер
3. Инженер по инфраструктуре
4. Аналитик
5. Дизайнер
6. Менеджер проекта и т.д.



**Перед оценкой проекта важно понимать полный объем работ, состав и квалификацию исполнителей задач проекта**

Task Name	Resource Names
<b>Work Change Form</b>	
<b>Общие доработки</b>	
Аналитика и прототипирование UI, согласование постановок	AL
Настройка доступности WCF группе новых пользователей	BE
Встраивание WCF модуля в интерфейс NBA	FE
Тестирование	QA
<b>Получение данных из источников и их отображение</b>	
Аналитика и прототипирование UI, согласование постановок	AL
Организация загрузки исходных данных в БД модуля	BE
Добавление 2 новых таблиц для хранения изменений	BE
Отображение в WCF данных из Client Order (таблица, фильтры, контролы)	FE
Отображение в WCF данных из CM Time Booking (таблица, фильтры, контролы)	FE
Тестирование	QA
<b>Обработка данных</b>	
Аналитика и прототипирование UI, согласование постановок	AL
Внесение пользователем данных через UI	FE
Загрузка данных из Excel	BE
Тестирование	QA
<b>Реализация эффектов обработки данных</b>	
Аналитика и прототипирование UI, согласование постановок	AL
Отображение отметок о наличии WCF корректировок в списке Client Order	BE
Отображение отметок ток о наличии WCF корректировок в позициях Client Order	BE
Отображение отметок ток о наличии WCF корректировок в CM Time Booking	BE
Выгрузка таблиц WCF в Excel	BE
Тестирование	QA
<b>Завершающий этап</b>	
Тестирование модуля целиком, отладка, проведение демо	QA
Актуализация документации	QA
Управление, координация команды	PM/2001



# Оценка проекта. Цель

1. Соотнесение объема работ с бюджетом проекта
2. Прогнозирование срока исполнения
3. Расчет потребности в ресурсах
4. Выявление рискованных задач
5. Понятный способ контроля выполнения проекта

# Несколько способов оценки

Метод оценки	Описание	Когда применять
Экспертные суждения	На основе мнений экспертов, обладающих опытом и знаниями в предметной области проекта. Недостатки: Субъективность, зависимость от опыта эксперта	Уникальные проекты, нужна высокая скорость оценки.
По аналогии	Оценка основана на данных предыдущих схожих проектов. Недостатки: низкая точность при различиях в условиях	Для типовых проектов
Мозговой штурм + покер планирования	Использование эмпирических правил и опыта команды. Недостатки: субъективность	Для типовых задач и высокой скорости оценки
Параметрическая	Выбираются параметры оцениваемого проекта и стоимость одной операции, строится мат. модель, вычисляющая оценку на основе этих данных. Недостатки: зависимость от качества данных и модели	Для повторяющихся задач
По трем точкам (PERT)	Используется три оценки: О – оптимистичная, П-пессимистичная, М-наиболее вероятная. Формула $E=(O+4M+П)/6$ , заложено стандартное отклонение и дисперсия Недостатки: требует детального анализа	Для высокоуровневой оценки
Метод Монте-Карло	Компьютерная симуляция, моделирует множество сценариев и рассчитывает вероятности их успеха Недостатки: сложность, нужны мощные инструменты	Нужна высокая точность, моделирование рисков

# Оценка проекта – единицы измерения

## Оценка в часах

Понятная бизнесу, легко вычисляется срок готовности.

Позволяет быстро оценить требуемые ресурсы, часто используется в больших проектах

## Оценка в условных единицах

Удобна для использования в agile командах для творческих задач, применяемые единицы и их истинный «размер» (очки сложности, условный размер задачи) зависят от опыта группы.

Требуется сравнение с предыдущими задачами для оценки новых. У менеджера задача оценки сроков и бюджета остается

Task Name	Work
Мобильный регистратор	2 120 hrs
Прототип: стиль Азимут, без связи с бэкендом	316 hrs
Дизайн	40 hrs
Разработка и согласование универсального стиля с параметризацией UI	40 hrs
Параметризация экранов уникальным стилем в зависимости от логина	96 hrs
Экран профиля пользователя	4 hrs
Поиск по брони, билету, рейсу, просмотр пассажиров	8 hrs
Экран бронирования	16 hrs
Домашний экран	4 hrs
Экран регистрации на рейс и печати посадочных	24 hrs
Багаж - добавление услуги	12 hrs
Выбор места - добавление услуги	8 hrs
Апгрейд - оплата добавленной услуги	8 hrs
Экран выписки услуг	4 hrs
Аннуляция и удаление услуг	8 hrs
Управление	180 hrs
Управление проектом	180 hrs
MVP: продажа багажа	1 548 hrs
v.1: + выбор места	0 hrs
Тестирование решения	160 hrs
Документация	40 hrs
Создание пользовательской документации	40 hrs
Поддержка	56 hrs

# План-график проекта

## Объединение всех предыдущих артефактов:

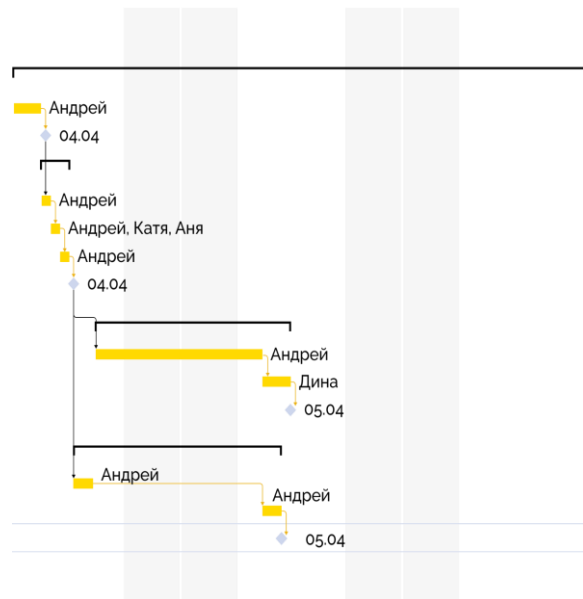
- Иерархическая структура работ
- Доступный ресурс команды
- Оценки задач

## Диаграмма Ганта –

горизонтальная временная шкала, на которой отображаются задачи, их длительность, последовательность выполнения и зависимость друг от друга

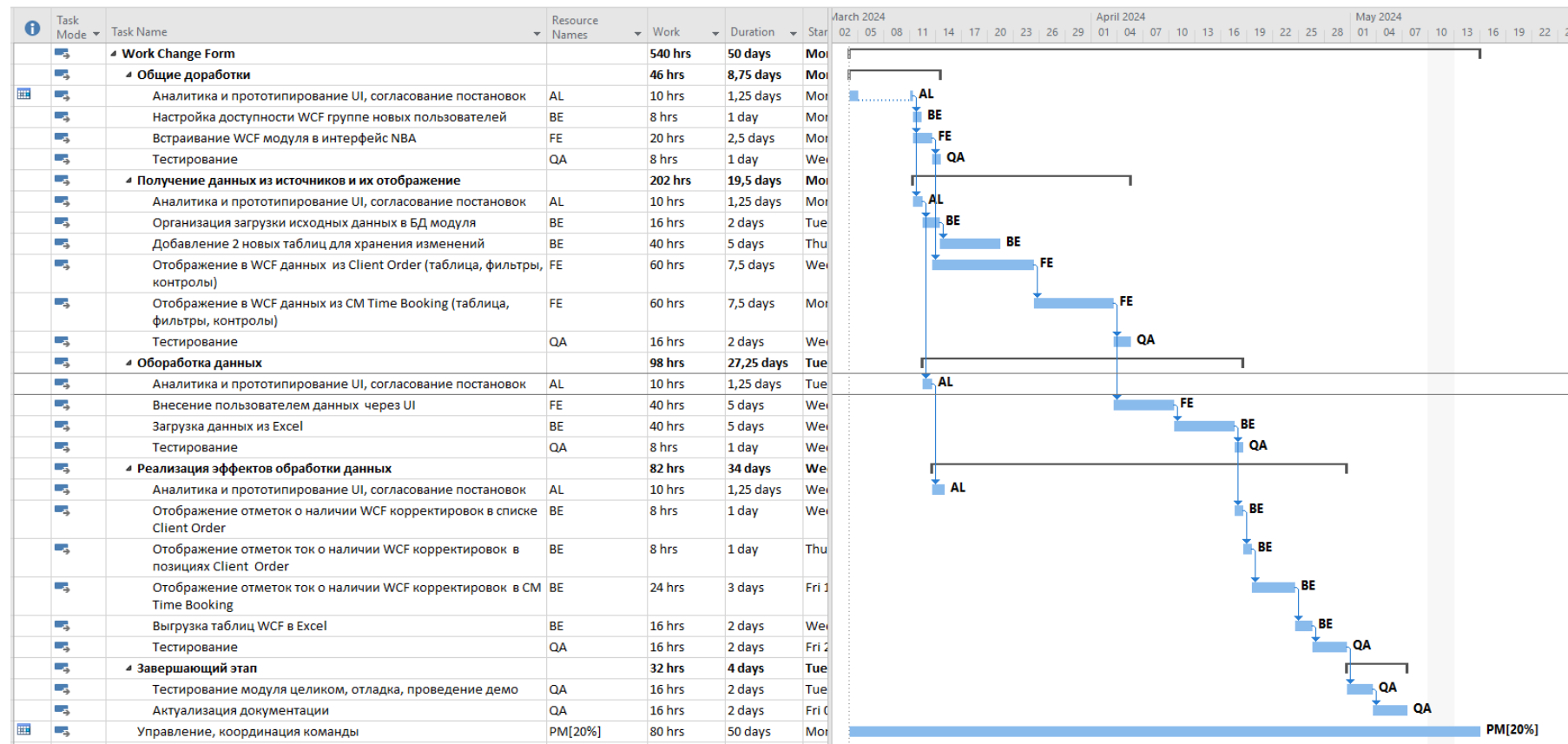
31

Позволяет дать ответ на вопрос о календарном сроке завершения проекта





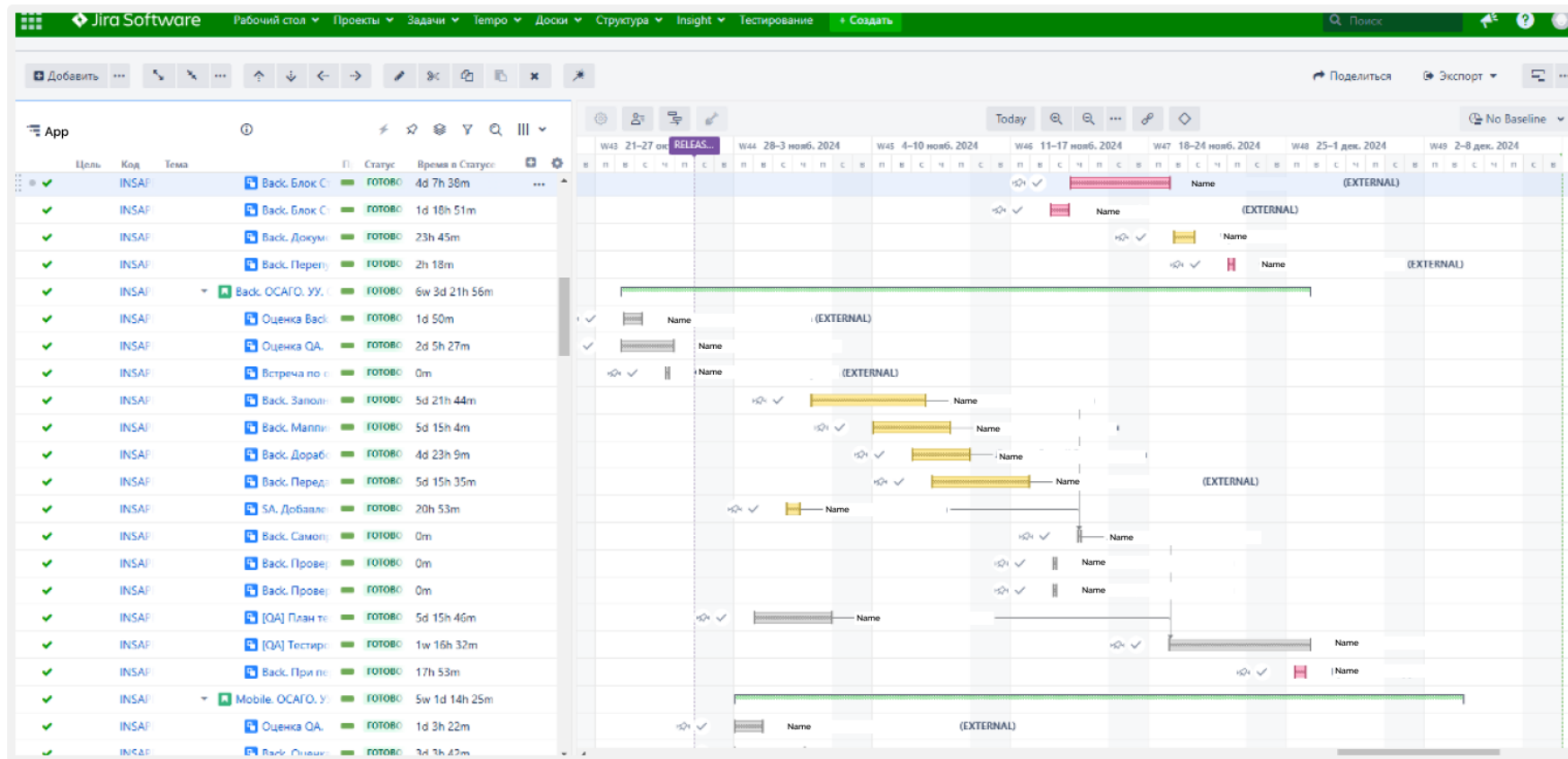
# Пример в Microsoft Project




# Особенности диаграммы Ганта

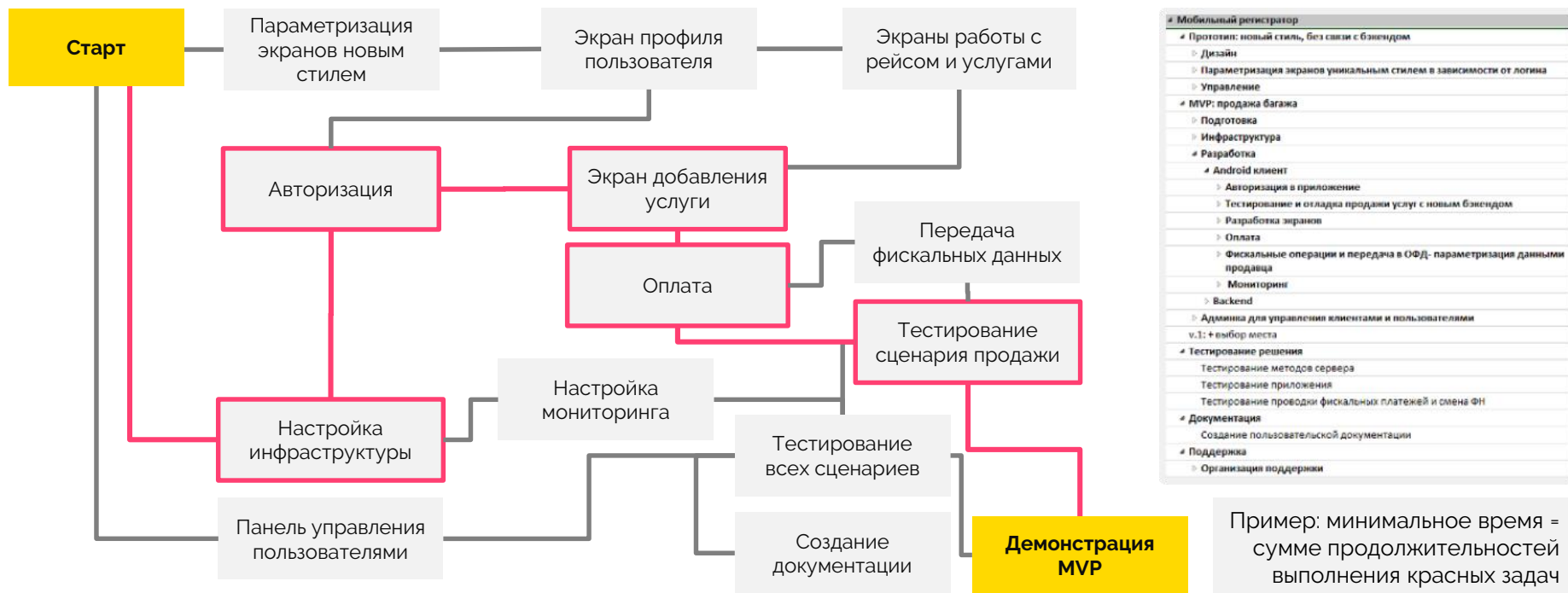
1. Широко известный формат
2. Удобно управлять зависимостями
3. Помогает выявлять отставания, планировать рабочую нагрузку на команду
4. Трудоемкость обновления.  
Устаревает сразу после начала выполнения проекта из-за изменения декомпозиции.  
Решение – объединять с трекером задач команды.
5. Требуется детального проектирования реализации
6. Применима к проектам среднего размера.  
Для маленьких избыточна, для очень крупных слишком трудоемка
7. Не отображает риски или вероятность выполнения задач

# Пример в Jira – автообновление диаграммы



# Метод критического пути

 Это метод планирования, используемый для определения самой длинной последовательности зависимых задач, которые определяют минимальное время завершения проекта



Мобильный регистратор	
■ Прототип: новый стиль, без связи с бэкендом	
▷ Дизайн	
▷ Параметризация экранов уникальным стилем в зависимости от логина	
▷ Управление	
■ MVP: продажа багажа	
▷ Подготовка	
▷ Инфраструктура	
■ Разработка	
▷ Android клиент	
▷ Авторизация в приложение	
▷ Тестирование и отладка продажи услуг с новым бэкендом	
▷ Разработка экранов	
▷ Оплата	
▷ Фискальные операции и передача в ОФД: параметризация данными продавца	
▷ Мониторинг	
▷ Backend	
▷ Админка для управления клиентами и пользователями	
v.1.1: + выбор места	
■ Тестирование решения	
▷ Тестирование методов сервера	
▷ Тестирование приложения	
▷ Тестирование проводки фискальных платежей и смена ФН	
■ Документация	
▷ Создание пользовательской документации	
■ Поддержка	
▷ Организация поддержки	

# Управление рисками

## Управление рисками –

процесс идентификации, анализа, оценки и реагирования на риски для минимизации их негативного воздействия на проект

Это проактивный подход к выявлению и устранению угроз для успешного завершения проекта

Процесс нужно повторять регулярно по мере хода проекта

## Стратегии:

- **Избегание** – не осуществлять часть проекта или работ, вызывающие этот риск
- **Сдерживание** – заложить в бюджет достаточно ресурсов, чтобы покрыть риск в случае его наступления
- **Ослабление** – принятие мер по снижению риска до его наступления, такие меры нужно сформировать и запланировать заранее
- **Принятие** – решить ничего не делать, взвесив все вводные



# Управление рисками

## Примеры рисков:

- Непредвиденные ошибки в новой библиотеке/технологии
- Уменьшение численности команды в середине проекта
- Выполнение задачи дольше прогнозируемой оценки
- Изменение требований в ходе проекта
- Большое количество ошибок при тестировании

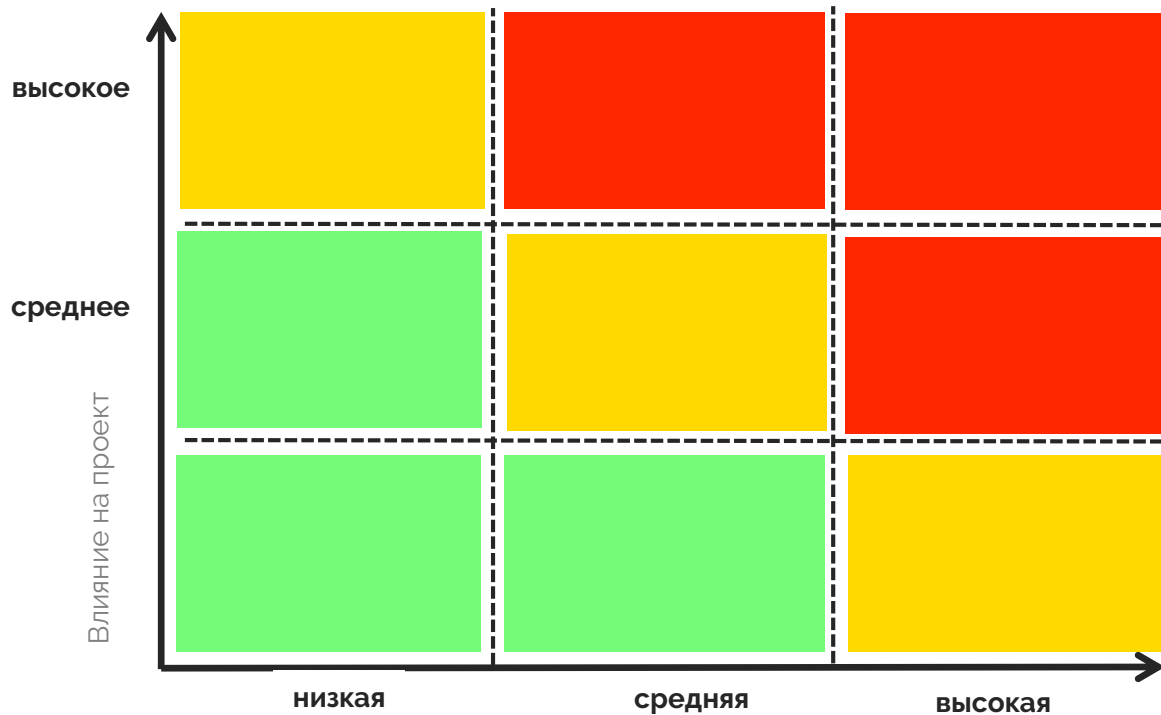
# Управление рисками. Шаги процесса

- 1. **Идентификация** – составить реестр рисков, используя экспертов или мозговой штурм
- 2. **Качественный анализ** – вычислить значимость риска, оценив вероятность наступления и степень влияния на проект
- 3. **Количественный анализ** – вычислить в деньгах влияние риска
- 4. **План реагирования** – способ реагирования на риск, план «Б», резервы, триггеры риска, ответственные
- 5. **Контроль** – вносить изменения в реестр рисков и актуализировать его

## Пример реестра рисков

Описание риска	Триггер возникновения	Последствия	Влияние	Вероятность	Значимость/приоритет	Мероприятия по минимизации	Стратегия реагирования	Ответственный
Потеря ключевого разработчика	Уход разработчика из команды	Смещение срока проекта	Высокое	Средняя	Высокая	Обучение других разработчиков Заложить резерв времени Хранить требования и комментарии в трекере задач	Перераспределить задачи	Тимлид

# Как оценить значимость риска



Вероятность возникновения

- ☐ **Критичные риски** – мероприятия по устранению включаем в план проекта, добавляем ресурсы
- ☐ **Средние** – формируем резервы и план «б», контролируем
- ☐ **Мало вероятные и не значимые** – контролируем





# Ход проекта – контроль

Контроль производится  
по **трем измерениям**

- Бюджет
- Срок
- Объем работ

## Контроль бюджета и объема работ

- **Бюджетная таблица** – это документ, который содержит информацию о планируемых и фактических тратах по каждой статье бюджета, а также о разнице между ними
- **Диаграмма Ганта** – позволяет сопоставить фактический ход выполнения работ с вехами и датами окончания работ по задачам, заложенными в диаграмму

## Контроль сроков

- Можно использовать метод контрольных точек. **Контрольная точка** — это конкретный проверяемый результат проекта, который должен появиться в установленный срок. При достижении контрольных точек производится сравнение плановых и фактических показателей.



## Контроль правильности смысла

- проводить регулярные демонстрации результата владельцу проекта и конечным пользователям

# Ход проекта – инструменты

- ✓ **Цель внедрения инструментов –**  
организация эффективного, прозрачного,  
гибкого процесса взаимодействия  
участников команды

Все инструменты можно сгруппировать  
по решаемым задачам и рассмотреть  
самые распространенные



# Ход проекта – инструменты

**Командные базы знаний,  
хранилища открытой  
информации**

Confluence, Notion, Wiki-  
платформы

1

**Инструменты детального  
планирования  
с поддержкой Gantt,  
ресурсных планов**

Microsoft Project, Asana,  
Team Gantt

2

**Инструменты  
концептуального  
моделирования  
и интерактивной работы**

Miro, Microsoft Whiteboard,  
Lucidchart

3

**Инструменты управления  
детальным составом работ  
(ИСР), требованиями и их  
изменениями**

Jira, TFS, Kaiten

4



# Ход проекта – инструменты

## Инструменты организации непрерывной работы

GitLab, CI/CD, Azure DevOps,  
Jenkins

5

## Инструменты коммуникаций

Microsoft Teams, Discord,  
Telegram, Rocket Chat

6

## Инструменты мониторинга работ

Grafana, PowerBI, Tableau

7

## Прочие инструменты

пакет офисных продуктов  
(документы, таблицы,  
презентации)

8

# Пример организации списка задач команды

+ New Work Item

Column Options

...

Спринт 24.10.2

Title	State	Rema...	Complete...	Original Es...	Assigne...	Tags
> ■ Unparented						
✓ SMRT-16652 MP BE: Передача фискальных данных на MP	● Done		26.5	6	Dmitry ...	DEV PROD UAT
⚙️ Вторичная оценка. MP BE: Передача фискальных данных на MP	● Done		1.25		Dmitry ...	
⚙️ Аналитика. MP BE: Передача фискальных данных на MP	● Done		8		Anton ...	
⚙️ Первичная оценка. MP BE: Передача фискальных данных на MP	● Done		3		Alexan...	
⚙️ Добавление FF. MP BE: Передача фискальных данных на MP	● Done		0.25		Alexan...	
📌 BE: имплементация передачи фискальных данных на MP	● Done		6	6	Alexan...	DEV FEATURE PROD qa ...
✓ SMRT-16489 MP: Необходимость FF SMRT-13409	● Done		19	8	Artem ...	UAT
⚙️ Первичная оценка. SMRT-16489 MP: Необходимость FF SMRT-13409	● Done		9		Artem ...	
📌 Обновление цены услуги Lounge	● Done		4	4	Artem ...	UAT
📌 Доработать функцию обновления Upgrade	● Done		3	4	Artem ...	UAT
🚫 Обновлять бронирование после прайсинга	● Done				Artem ...	UAT
🚫 BE: Ошибка при указании всех типов услуг	● Done				Andrey...	DEV FEATURE PROD qa ...
> ✓ SMRT-16357 MP: Отображение скидок по услуге Бизнес-зал	● Done		23.3	25	Olga A...	UAT
✓ SMRT-16451 MP: Подготовка релиза сборки 7.12.0	● Done		34		svc-tfs-...	PROD UAT
> ✓ SMRT-16312 MP: МПК/МПБ. Отображение скидки	● Done		99.7	72	Olga A...	feature-flag-on UAT
> ✓ SMRT-12392 MP: СНБ отображение скидки	● Done		42.55	28	Olga A...	UAT
> ✓ SMRT-16146 MP: Передача платежной информации	● In Test		95.25	62	Artem ...	PROD UAT
> ✓ SMRT-14253 Мобильный Регистратор - Обновление документации	● Done		47.25	5	Anton ...	
> ✓ SMRT-15680 Tech Story MR: Передача чувствительной информации в открытом виде	● Ready for Test		21	6	Artem ...	UAT

# Управление изменениями

## Управление изменениями –

процесс, направленный на проведение и координацию изменений в проекте. Потребность в изменениях возникает в ответ на меняющиеся внутренние или внешние факторы.

**Цель** изменений – сохранение границ проекта и исполнение обязательств



## Примеры

- обновление бизнес-процессов, внедрение новых регламентов
- создание новых должностей, которое приводит к пересмотру устоявшихся процессов
- внедрение новых IT-продуктов для управления: CRM-систем, ERP-систем и тому подобного
- создание новых подразделений
- замена оборудования
- обновление технологий производства

# Управление изменениями

ADKAR

Модель разработана CEO Prosci Change Джеффом Хиаатом. В центре этой концепции — сотрудники, которых касаются изменения. ADKAR представляет собой набор **из пяти этапов**, которые последовательно должен пройти каждый работник

A

## Awareness

Осознание  
необходимости  
перемен

D

## Desire

Желание поддержать  
изменения  
и участвовать в них

K

## Knowledge

Знание, что делать во  
время и после  
изменений

A

## Ability

Способность  
претворить  
изменения в жизнь

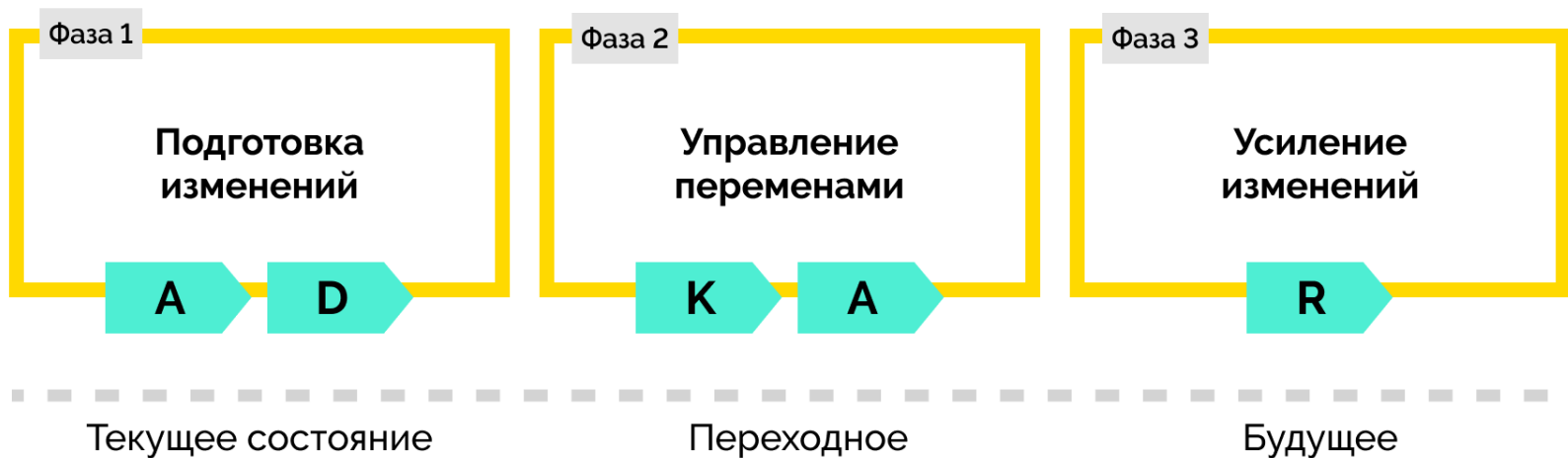
R

## Reinforcement

Закрепление  
результатов  
изменений

# 3-х фазный процесс управления изменениями

ADKAR





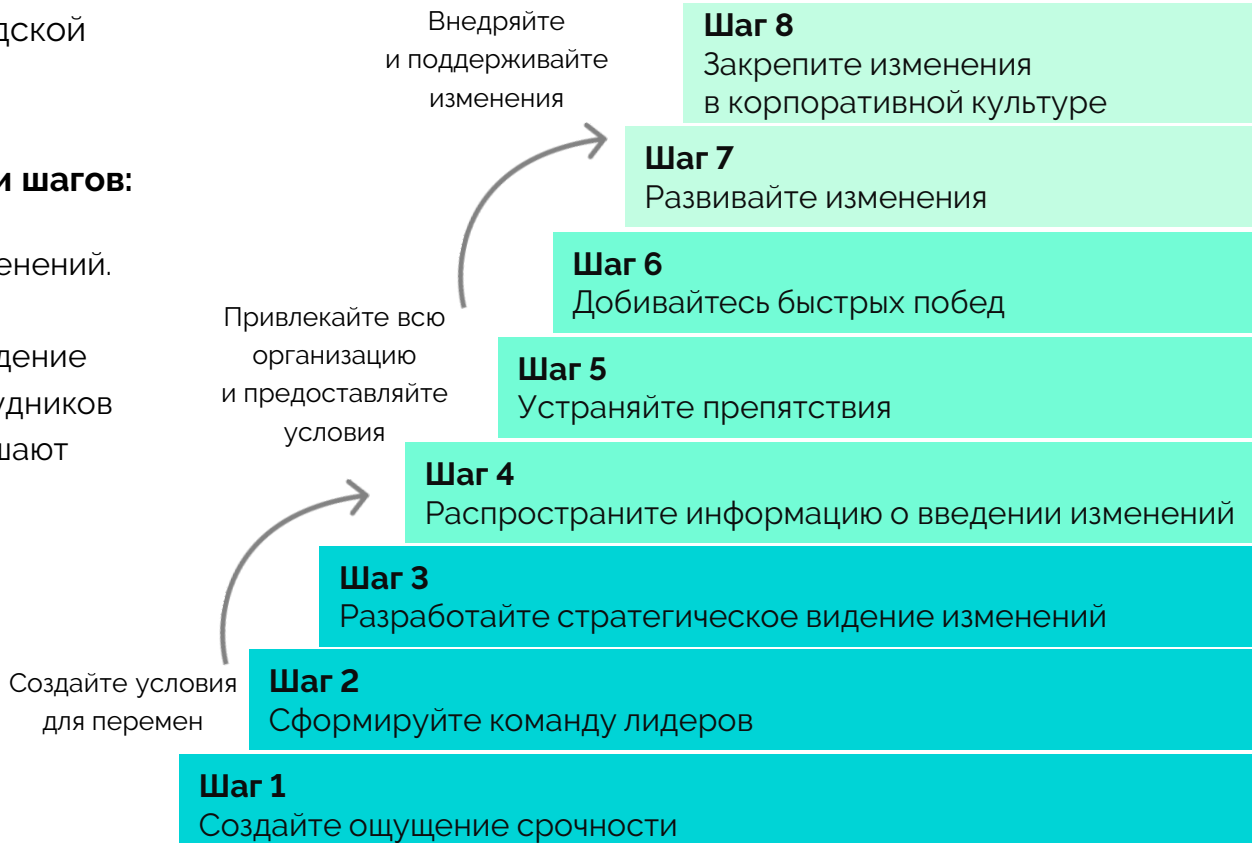
# Управление изменениями

## Модель Коттера

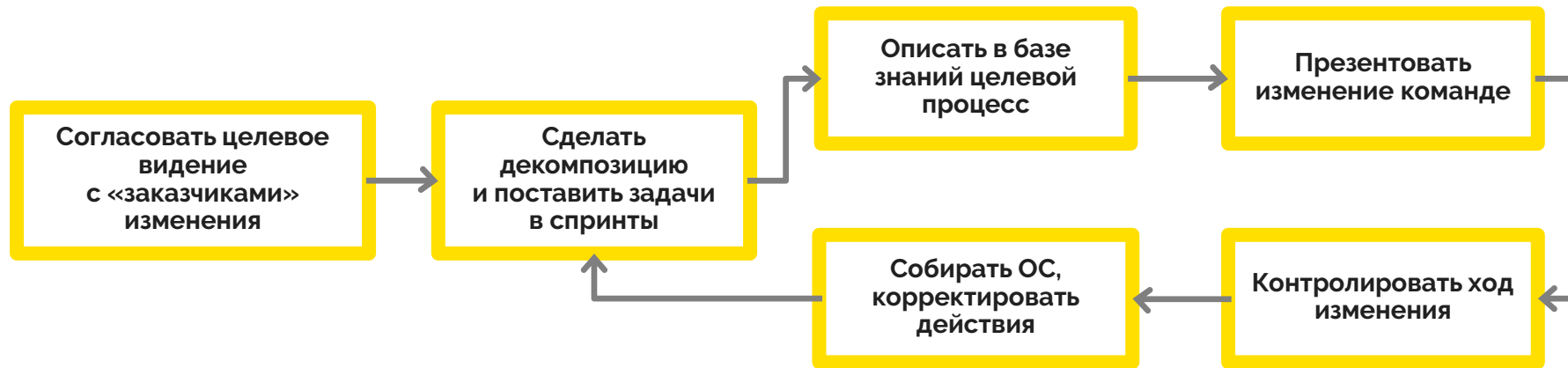
Разработана профессором Гарвардской школы бизнеса Джоном Коттером.

Модель Коттера — это **последовательность из восьми шагов**:

- Создать ощущение срочности изменений.
- Создать руководящую коалицию
- Сформировать стратегическое видение
- Собрать сторонников среди сотрудников
- Преодолеть барьеры, которые мешают изменениям
- Добиваться краткосрочных побед и отмечать первые успехи
- Постепенно наращивать темп внедрения изменений
- Закрепить результат



# Жизненный цикл изменения



## ДО начала проведения изменений:

1. Выбрать ответственного изменений.
2. Определить целевое состояние
3. Замерить текущие метрики

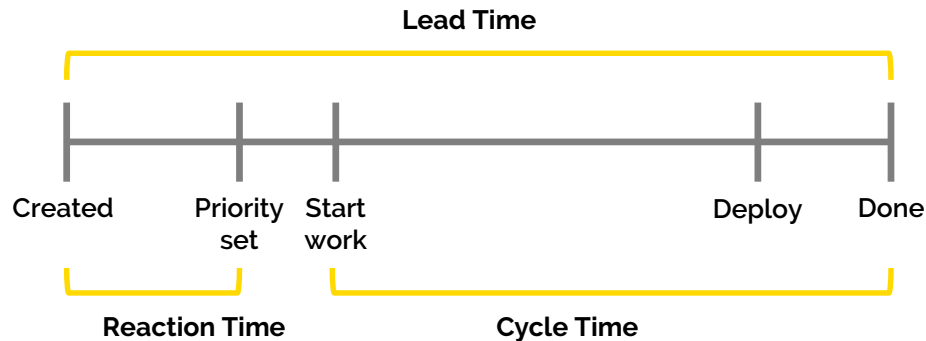
## ПОСЛЕ проведения изменений:

1. Замерить метрики и сравнить с начальными.
2. Принять решение о закрытии задачи
3. Уведомить заинтересованных о результатах

# Метрики команды разработки

Позволяют оценить производительность команды, контролировать прогресс и обнаруживать узкие места в процессе разработки.

В каждом фреймворке управления набор метрик будет немного отличаться, но ключевых несколько



- **Объем незавершенного производства (или work in progress)** – количество задач, потоков разработки, которое находится в работе у команды в один момент времени. Для повышения эффективности значение нужно сводить к минимальному.
- **Скорость (velocity)** – количество задач, завершенных за спринт.
- **Time to market** – время от начала разработки до релиза задачи/продукта.
- **Cycle time** (время выполнения) – время от поступления задачи в работу команды, до ее завершения
- **Lead Time** – время от создания задачи в бэклоге, до ее завершения.

# Обеспечение качества

Процесс, направленный на обеспечение соответствия продукта требованиям и ожиданиями заказчика

## Планирование

- Определение стандартов качества
- Разработка планов контроля
- Определение метрик качества
- Определение процесса обеспечения качества

## Обеспечение

- Выявление и устранение дефектов через регулярное тестирование и аудит результатов работы

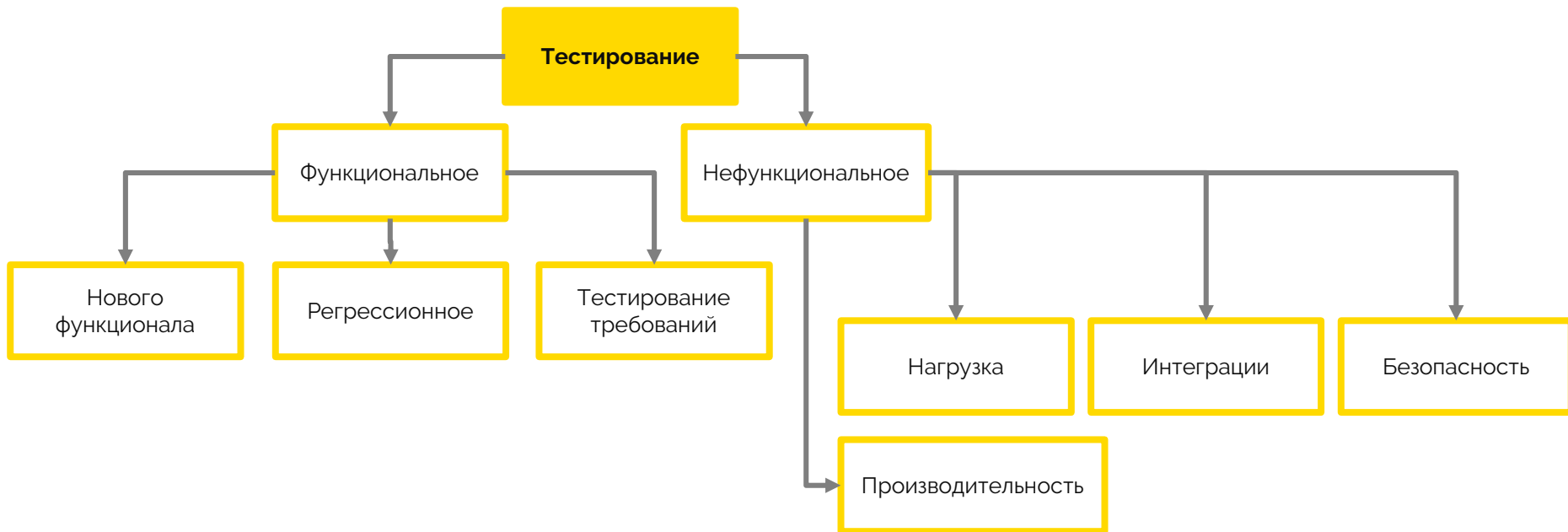
## Контроль

- Проверка результатов тестирования
- Анализ метрик
- Корректировка процесса



**Прослеживаемость качества** – способность отслеживать, документировать и подтверждать соответствие каждого результата установленным требованиям на всех этапах проекта

# Обеспечение качества



Автоматизация тестирования

# Стратегия тестирования

## Стратегия тестирования —

документ, описывающий подход и принцип тестирования ПО в рамках проекта.

Она определяет цели, области тестирования, типы тестов, инструменты, роли и ответственность команды, а также критерии успешного завершения тестирования

## Цели стратегии

- **Обеспечить качество продукта:** выявить ошибки и дефекты на ранних стадиях разработки
- **Минимизировать риски:** снизить вероятность критических ошибок в продакшн
- **Обозначить границы тестирования:** определить что будет тестироваться, а что нет
- **Эффективное использование ресурсов:** оптимизировать усилия команды за счет организации процесса, автоматизации

# Инструменты тестирования

Этап производства	Действия QA инженера	Инструмент
Инициация проекта	Анализ требований и спецификаций проекта Определение критериев качества Планирование стратегии тестирования	Confluence, Jira, TestRail, Docs, Trello
Проектирование	Участие в проектировании архитектуры тестов Создание тестовой документации (тест-кейсы) Анализ рисков	Draw.io, Miro, TestRail, Zephyr, Excel
Разработка	Подготовка тестовых данных Написание автотестов Ручное тестирование отдельных функций	SQL, Selenium, Postman, Junit, Browser DevTools, Fiddler
Тестирование	Функциональное, интеграционное, регрессионное, нагрузочное тестирование Отчет об ошибках Верификация исправлений	Jira, TestRail, Selenium, SoapUI, Jenkins, TestNG, Jmeter, Gatling, Bugzilla
Деплой и релиз	Smoke тестирование Проверка критериев приемки Валидация и верификация релиза	Confluence, Excel, Postman, Jenkins, Jira, TestRail
Сопровождение	Мониторинг ошибок в продакшн Тестирование обновлений и патчей Поддержка пользователей (ситуативно)	Kibana, Grafana, Service Desk systems

# Чек-листы для тестирования функционала

← MP: Чеклисты к задачам ☆ ⋮

Oct 17 - Oct 24 Past

0% run. [View report](#)

Test Suites

334143 : SMRT-13713 MP: Печать поса... ..

334799 : SMRT-13732 MP: Журнал ош... (...

286931 : SMRT-12488 MP: Экран центр... ..

343905 : SMRT-13990 MP: В чеке прихо... ..

342462 : SMRT-13940 MP: Обработка о... ..

341597 : SMRT-13908 MP: Обработка о... ..

352672 : SMRT-14280 MP: изменить ото... ..

346432 : SMRT-14075 MP: Редактирова... ..

350035 : SMRT-14178 MP: Подтвержде... ..

354952 : SMRT-14386 MP: работа с част... ..

360118 : SMRT-14636 MP: аннуляция Е... ..

358068 : SMRT-14525 MP: Частичная ан... ..

366697 : SMRT-15063 MP: Передача в С... ..

416903 : SMRT-16312 MP: МПК/МПБ... ..

284229 : SMRT-12392 MP: СНБ отобра... ..

313992 : SMRT-13133 MP: Оплата UPG... ..

315597 : SMRT-13167 MP: Не выполнят... ..

189070 : SMRT-10346 MP: предварител... ..

193858 : SMRT-10507 MP: Регистрация... ..

192580 : SMRT-10473 MP: Регистрация... ..

284229 : SMRT-12392 MP: СНБ отображение скидки (ID: 432490)

ⓘ Help

Close Grid

Define Execute Chart Offline test execution

IDTitleStep ActionStep Expected ResultAssigned ToState

432491

Отображение скидки для багажа

Предусловие: Включить FF SMRT-12392 MP: СНБ отображение скидки

Предусловие: создано трансферное бронирование для ADT+CHD+INF+INS с билетом, тариф basic

Отображение скидки СНБ

Зачитать бронирование в MP

Перейти к добавлению багажа

Выполнить прайсинг багажа для CHD на 2 сегмента

На плашке СНБ показано:  
-стоимость со скидкой в валюте (поле convertedPriceWithDiscount.amount)  
-стоимость со скидкой в милях (поле priceMiles)  
-полная стоимость в валюте (convertedPriceWithDiscount.amount+discountFloat)  
-полная стоимость в милях (priceMiles+discountMiles)  
(пример во вложении)

Рассчитать такой же багаж на 2 сегмента в ST, сравнить суммы в MP и ST

Кнопка "Оплатить" не активна

Добавить багаж

На свернутой плашке СНБ показано:  
-стоимость со скидкой в валюте (поле convertedPriceWithDiscount.amount)  
-стоимость со скидкой в милях (поле priceMiles)

Maxim Marchenko

Ready



# Внедрение продукта

## Внедрение —

это финальная фаза, на которой результаты проекта передаются заказчику или конечным пользователям.

Менеджер проекта планирует и координирует мероприятия, чтобы гарантировать успешное завершение проекта

# Внедрение продукта



# Метрики

## 1. Бизнес-показатели

- Конверсия
- Объем продаж
- Количество пользователей и т.д.

## 2. Системные

- Нагрузка на инфраструктуру
- Производительность сервисов и т.п.

## 3. Технологические показатели

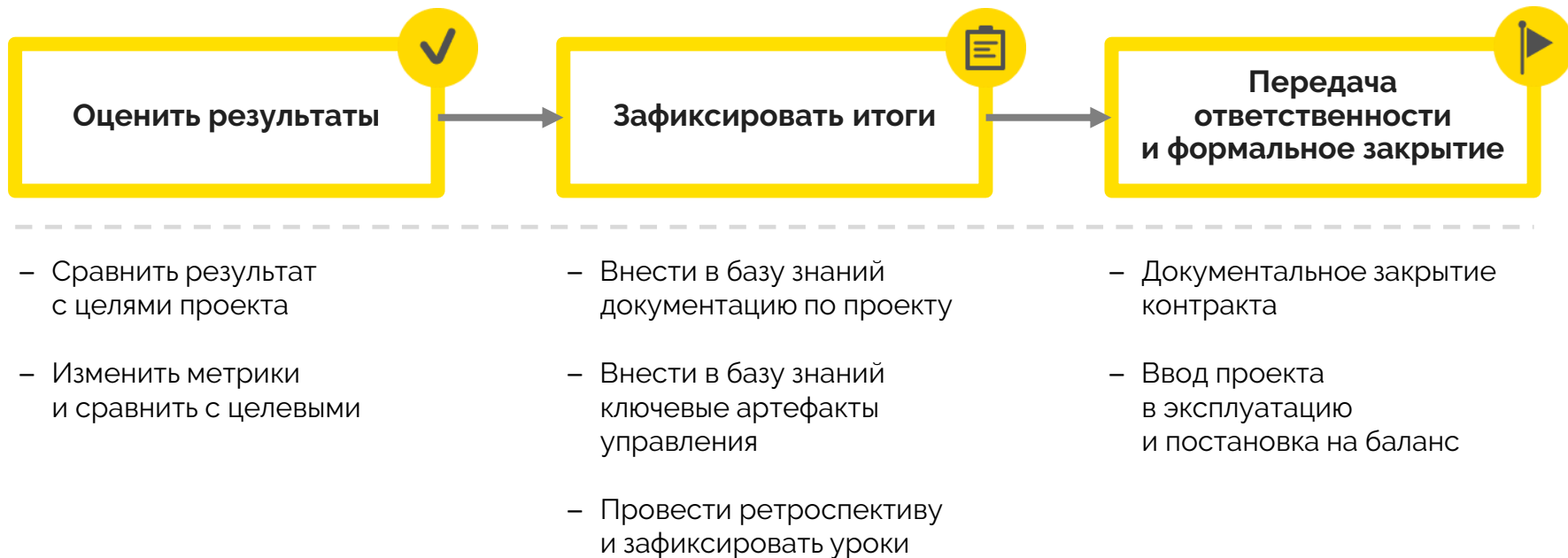
- действия пользователей
- вызовы сервера
- Внутренние операции системы



# Бизнес-метрики приложения опросника о качестве обслуживания

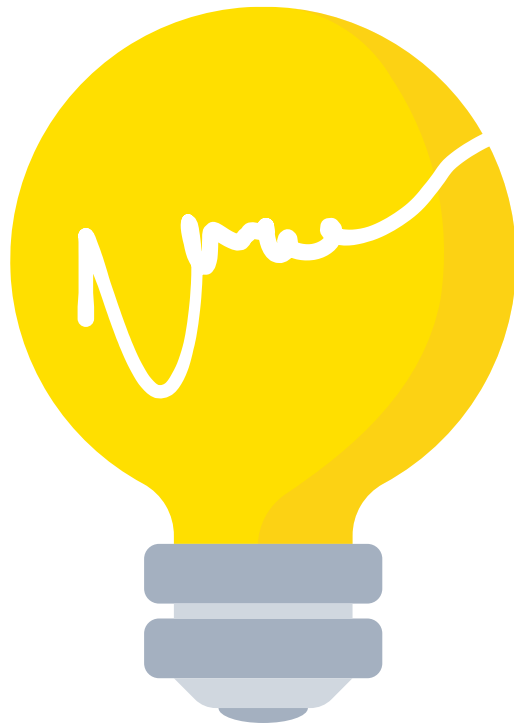


# Завершение проекта



# Ключевые мысли

1. Планирование помогает управлять проектом, но не гарантирует успешное его исполнение
2. Оценка и планирование должно учитывать состав команды и квалификацию участников
3. Применение инструментов для визуализации работы и совместной деятельности повышает эффективность команды
4. Управление изменениями – важнейший навык руководителя проекта для обеспечения гибкости и устойчивости к рискам
5. Прозрачность и постоянный контакт с владельцем продукта и пользователями позволяет сделать полезный и востребованный продукт



---

**Вопросы?**



**N\***

**Спасибо  
за внимание**