



ВЫСШАЯ ШКОЛА БИЗНЕСА
ЦЕНТР КОРПОРАТИВНОГО ОБУЧЕНИЯ



TMX
ИНЖИНИРИНГ

#ШКОЛАНОВАТОРОВ

УПРАВЛЕНИЕ ПРОЕКТАМИ: ГИБКИЕ ПОДХОДЫ К УПРАВЛЕНИЮ

04.02 - 11.02 | Александр Колесников

ПРОГРАММА КУРСА



- Предпосылки возникновения
- Симуляция One Piece Flow
- Основные подходы
- Работа с инструментами
- Модели сложности, сходства и различия подходов, границы применимости
- Оценки и приоритеты
- Целеполагание в гибких подходах

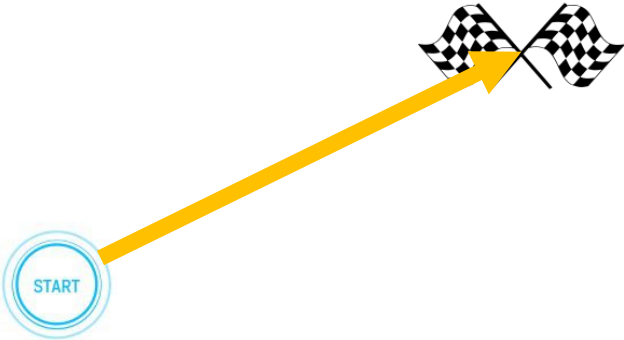
ПРЕДПОСЫЛКИ



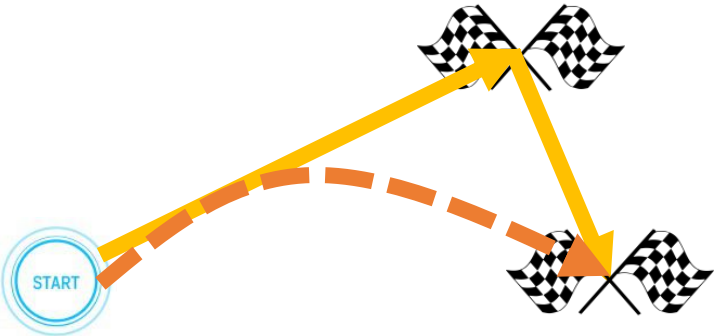
ВСЕ ПОСТОЯННО МЕНЯЕТСЯ



Идеальный мир



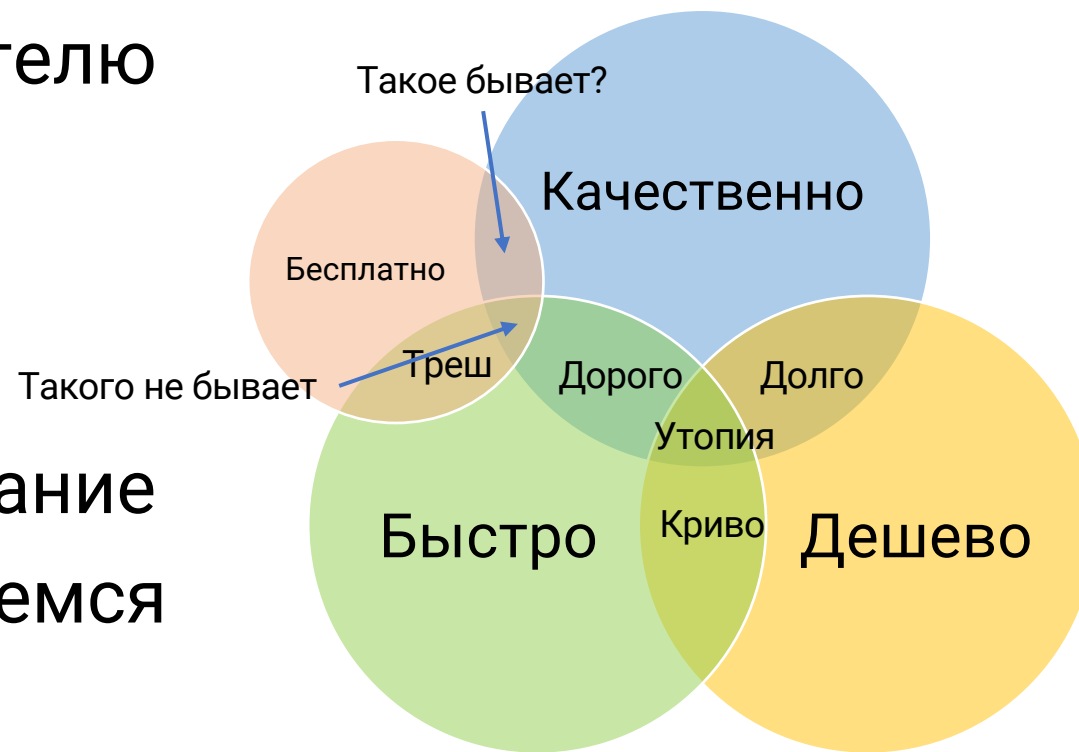
Реальный мир



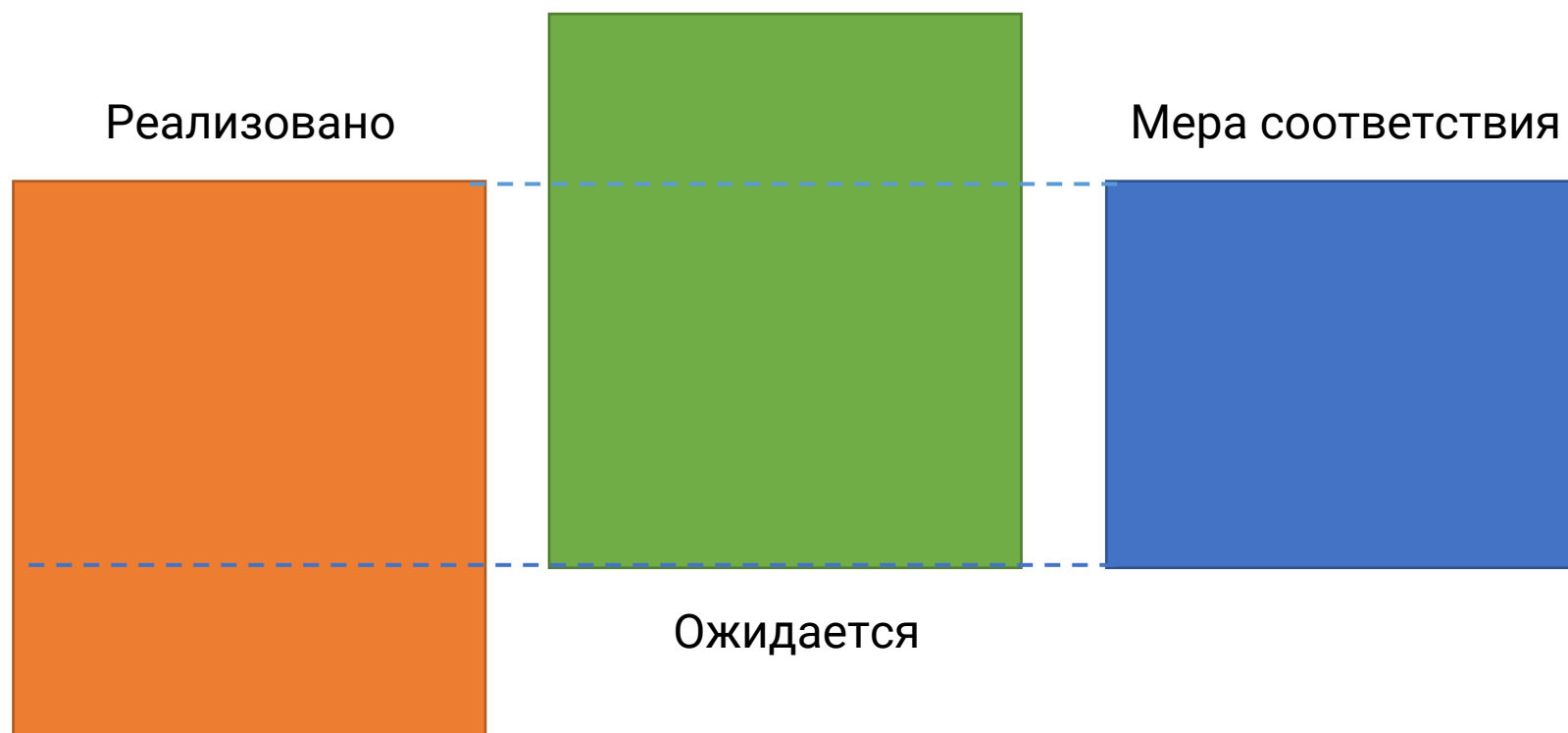
ЧТО НУЖНО СДЕЛАТЬ?



- Организовать соответствующие процессы
- Понять, что нужно потребителю
- Вписаться в бюджет
- Вписаться в сроки
- Понимать происходящее
- И уметь донести это понимание
- И все это в быстроменяющемся мире



НА СКОЛЬКО КАЧЕСТВЕННО МЫ МОЖЕМ УДОВЛЕТВОРИТЬ ПОТРЕБНОСТИ?



НА СКОЛЬКО КАЧЕСТВЕННО МЫ МОЖЕМ УДОВЛЕТВОРИТЬ ПОТРЕБНОСТИ?



ЧТО ТАКОЕ КАЧЕСТВО?



Это степень
удовлетворенности
потребителя и соответствия
свойств продукта его
ожиданиям. (ISO 9000)



Пирамида качества Гойко Аджича

ДЛЯ ЧЕГО НУЖНЫ ГИБКИЕ МЕТОДОЛОГИИ УПРАВЛЕНИЯ ПРОЕКТАМИ (ГМУП)



От осознания проблем в управленческих процессах, через стандартизацию, оптимизацию, проектный подход, статистическое управление и т.д., мы подходим к управлению продуктами и созданию в них ценности

ГМУП – это о том, как делать продукты, создавать дополнительную ценность

AGILE И LEAN



Эти два понятия часто идут рядом, вплоть до смешения

Истоки у них разные

В дальнейшем, говоря про Agile или Гибкие методологии, будет подразумеваться и Agile, и Lean

AGILE



Agile [эджáйл] – перевод с английского: «быстрый», «расторопный», «сообразительный».

Можно определить, как набор подходов к разработке продуктов, ориентированный на параллельное, адаптивное и итеративное выполнение различных этапов жизненного цикла проекта в результате постоянного взаимодействия самоорганизующихся кроссфункциональных рабочих групп.

Примером Agile подхода является Scrum

LEAN



Lean [лин] – это бережливое производство. Это набор принципов, относящихся к ориентированности на клиента, на качество, на скорость, на постоянное стремление к устранению всех видов потерь.

Примером Lean подхода является Kanban

ДРУГИМИ СЛОВАМИ



Agile – более революционный подход, который дает возможность получать дополнительную ценность (added value). Ориентирован на продукт.

Lean – более эволюционный подход, который дает возможность сокращать издержки (reduced debt). Ориентирован на процесс.

ПРОТИВОСТОЯНИЕ ТРАДИЦИОННЫХ И ГИБКИХ ПОДХОДОВ



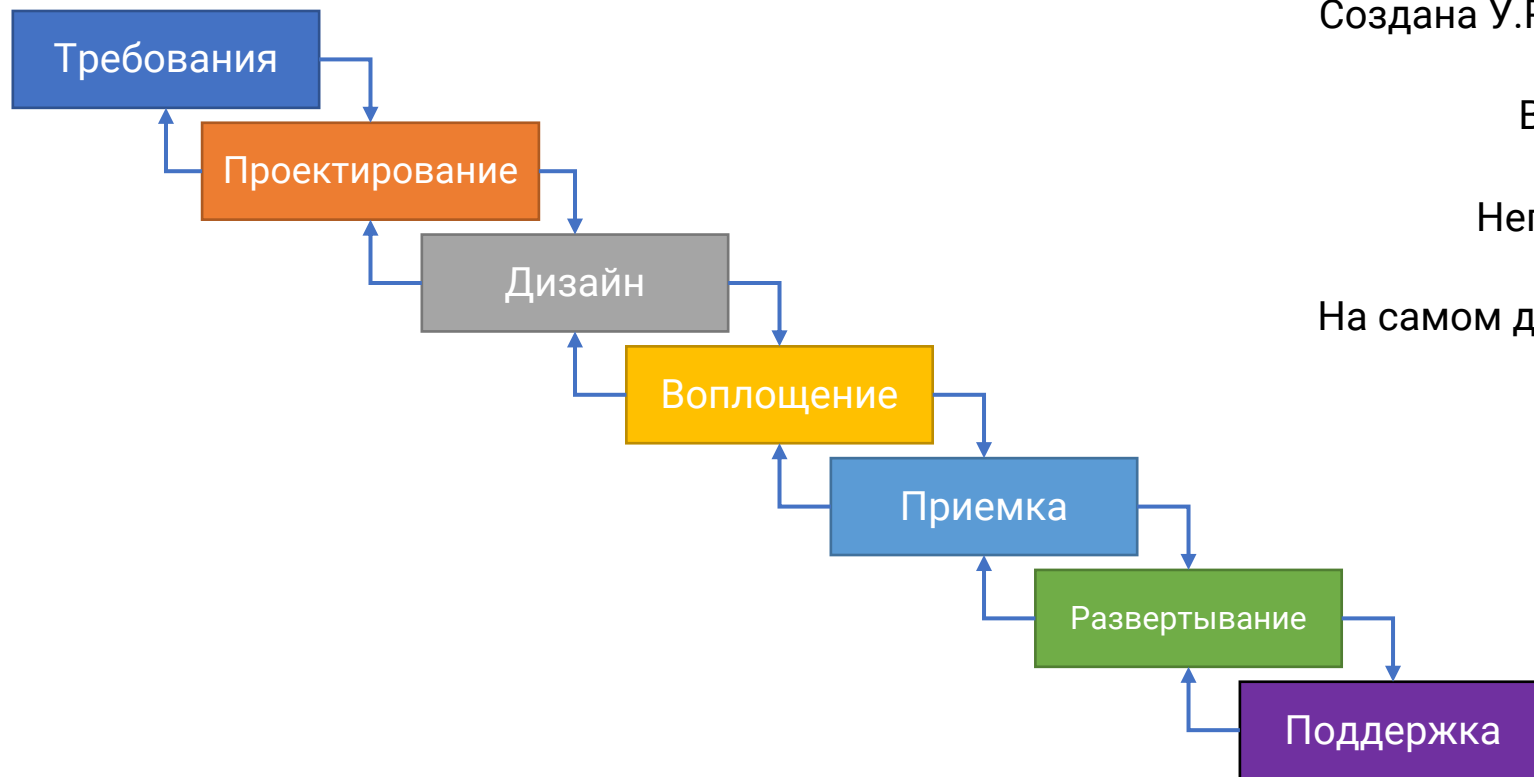
Это миф

Это ответная реакция на перегруженность на неразумное использование существующих подходов

Это эволюционное развитие традиционных подходов

Это органичное дополнение традиционных подходов управления проектами

КАСКАДНАЯ МОДЕЛЬ



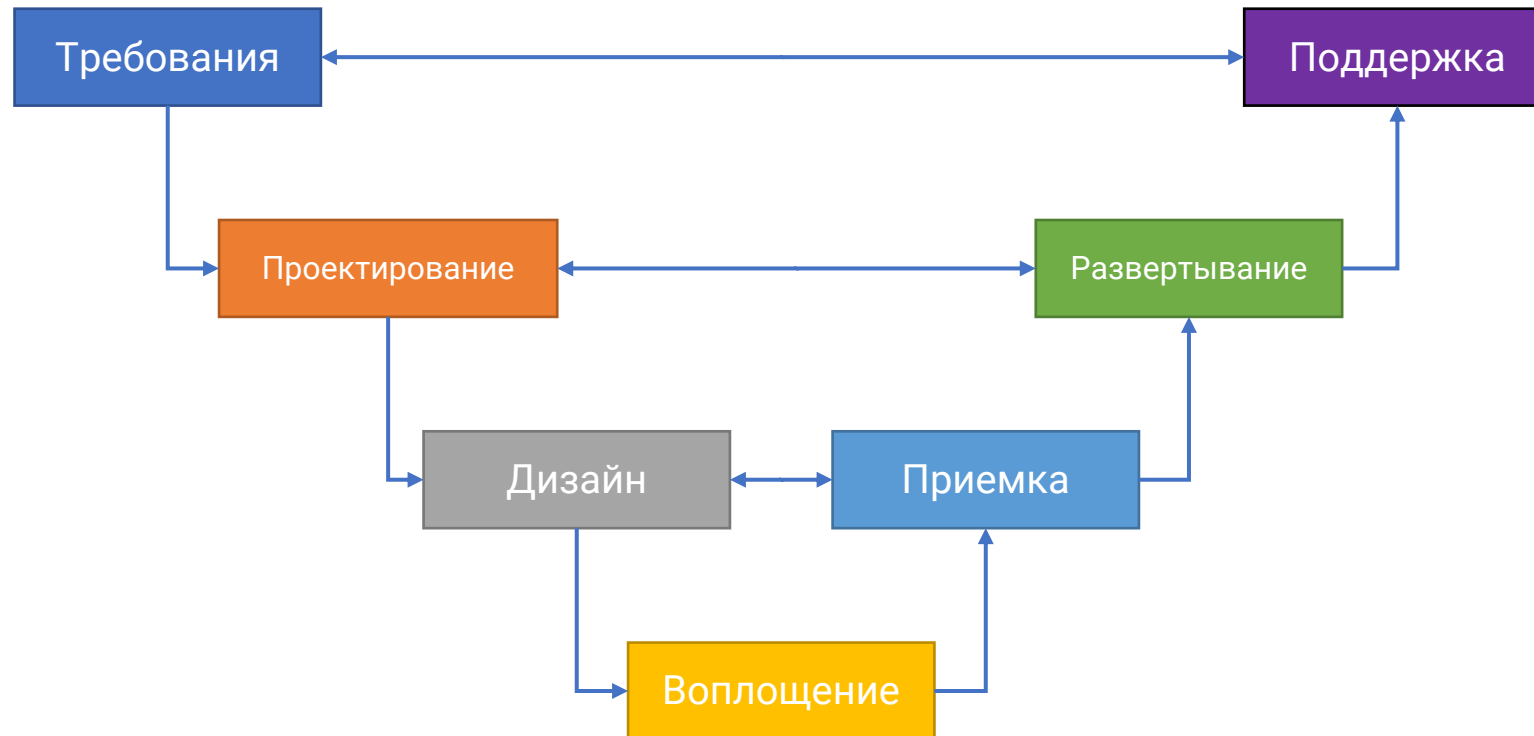
Создана У.Ройсом в 1970 году

Всеми критикуемая

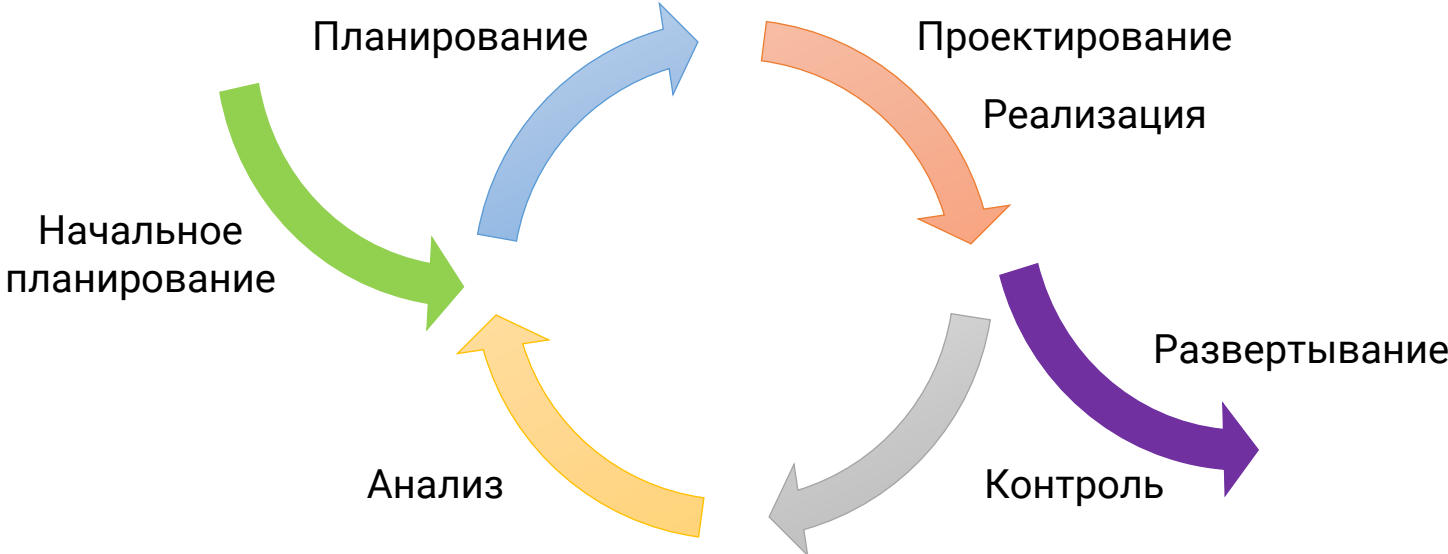
Неправильно понятая

На самом деле он имел ввиду...

V-МОДЕЛЬ

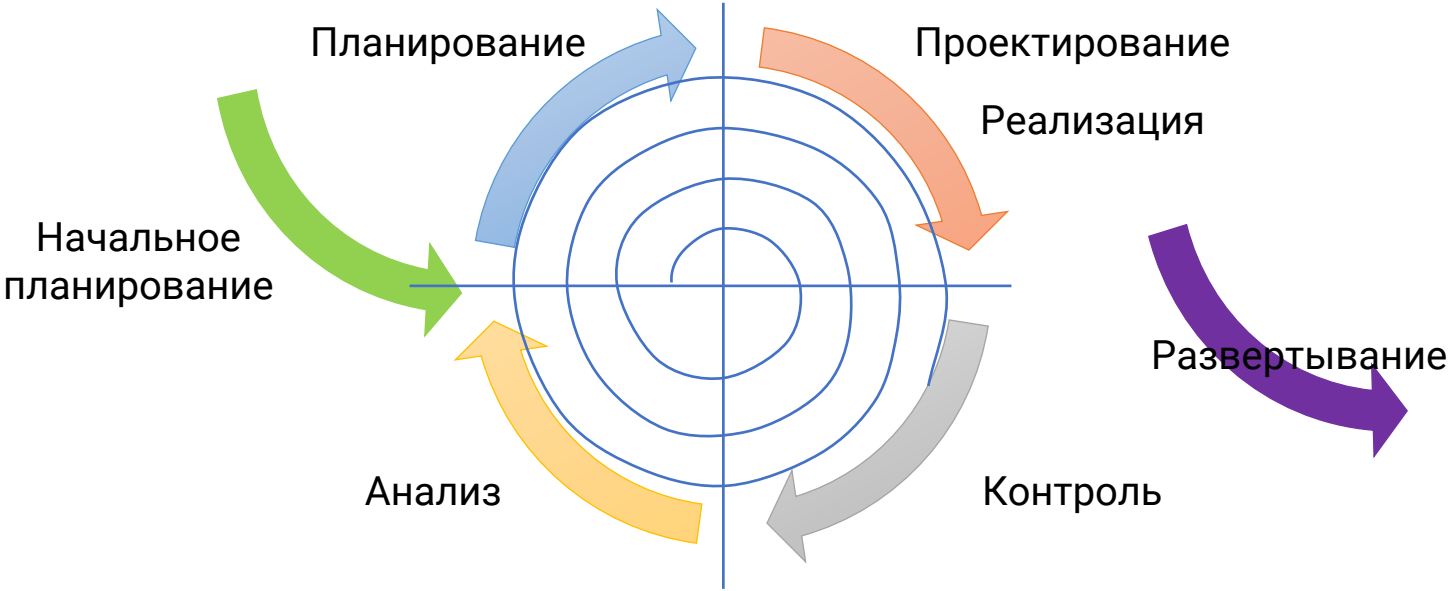


ИТЕРАТИВНАЯ МОДЕЛЬ

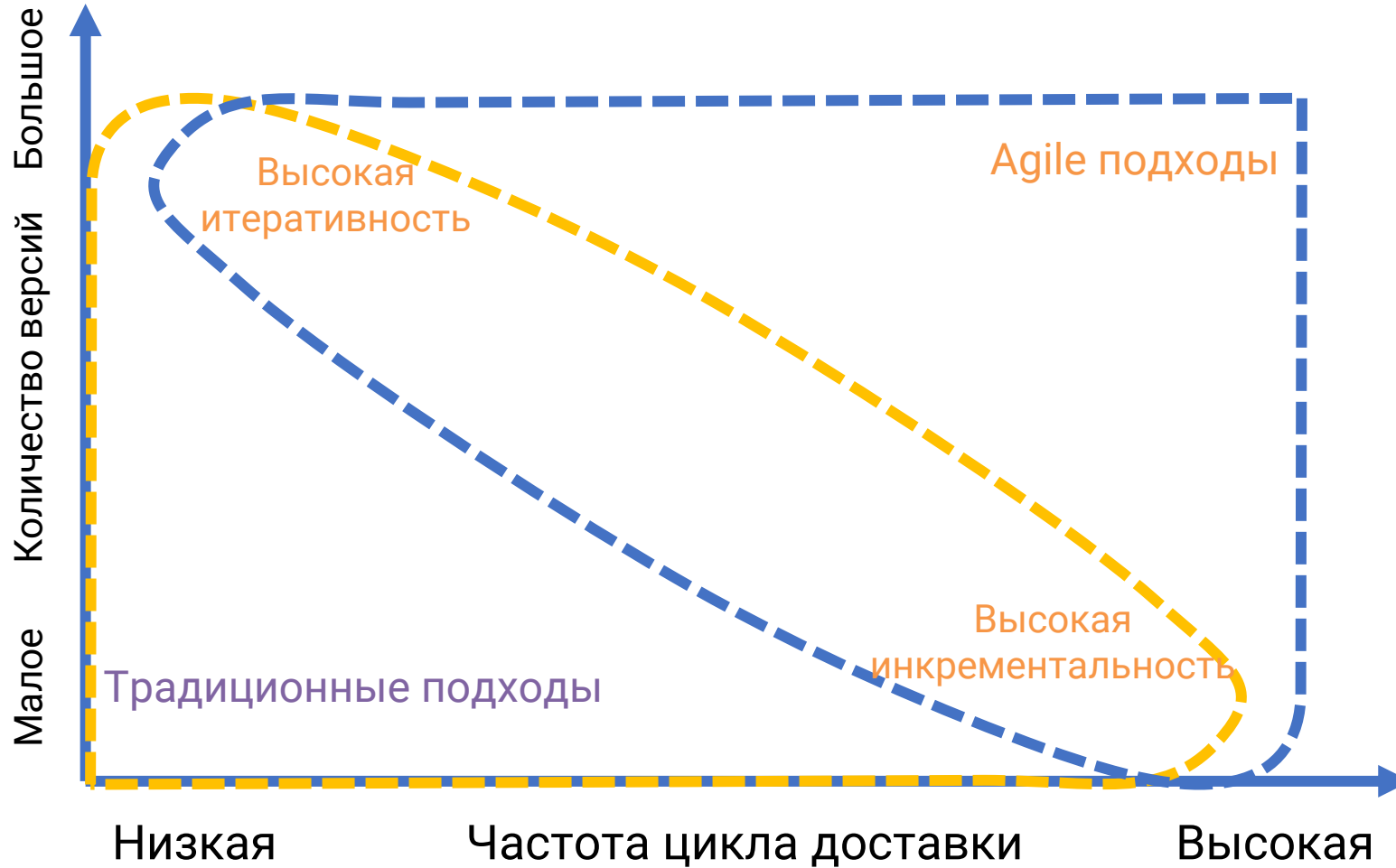


Цикл Деминга (PDCA => OODA)

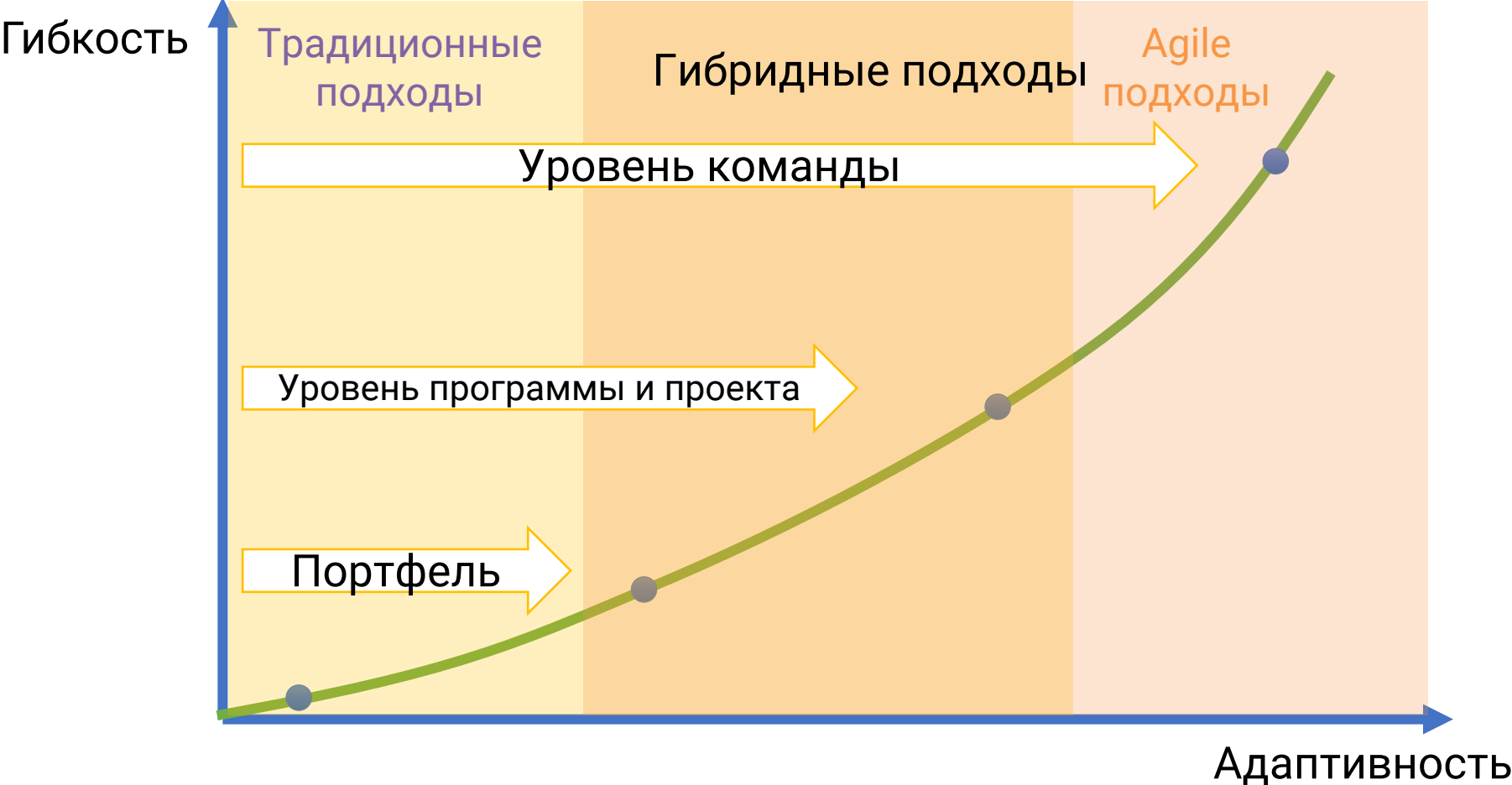
СПИРАЛЬНАЯ МОДЕЛЬ



КАКОВА ОБЛАСТЬ ПРИМЕНЕНИЯ



НА КАКОМ УРОВНЕ ПРИМЕНЯТЬ (ПРИМЕР)



СИМУЛЯЦИЯ ONE PIECE FLOW



ИГРА: ONE PIECE FLOW SIMULATION 4.1



- 20 монет
- 4 работника
- Измеряем время работы каждого работника
- Измеряем время поступления первой партии на рынок
- Измеряем общее время работы
- Все 20 монет
- По 10 монет
- По 5 монет
- По 1 монете

ИГРА: ONE PIECE FLOW SIMULATION 4.2

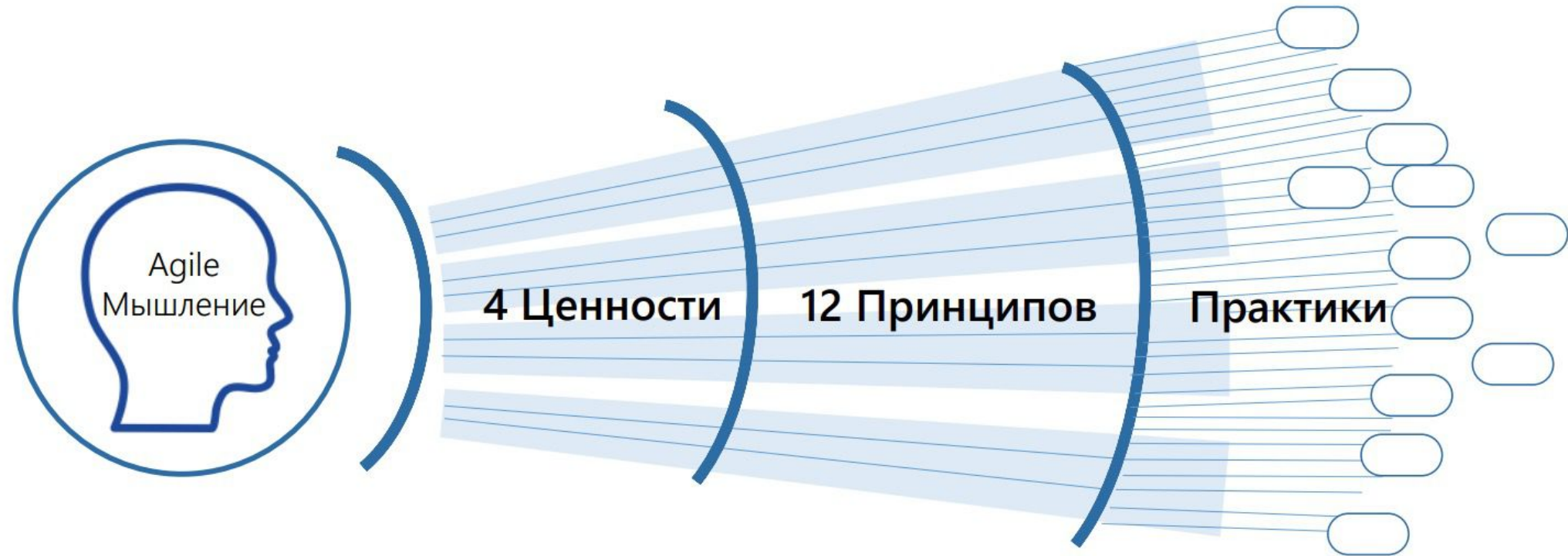


- Один сотрудник
- 5 менеджеров
- Сотрудник должен написать имя каждого
- Измеряем время написания первого имени
- Измеряем время все работы
- Все действуют наперебой
- Сотрудник пишет по одной букве каждого имени
- Сотрудник по очереди пишет полностью имя каждого

AGILE ПОДХОД НА ПРИМЕРЕ SCRUM



AGILE – ЭТО ОБРАЗ МЫШЛЕНИЯ



<http://agilemanifesto.org/iso/ru/manifesto.html>
<http://agilemanifesto.org/iso/ru/principles.html>

AGILE-МАНИФЕСТ (2001 Г.)



Мы постоянно открываем для себя более совершенные методы разработки программного обеспечения, занимаясь разработкой непосредственно и помогая в этом другим. Благодаря проделанной работе мы смогли осознать, что:

Люди и взаимодействие важнее **процессов и инструментов**
Работающий продукт важнее **исчерпывающей документации**
Сотрудничество с заказчиком важнее **согласования условий контракта**
Готовность к изменениям важнее **следования первоначальному плану**

То есть, не отрицая важности того, что справа, мы всё-таки больше ценим то, что слева.

НЕ AGILE



Если мы говорим о процессах, документах, встречах, организационных моментах, то это **НЕ** Agile

Если мы говорим о людях, о фильтрах информации на вход и выход, об «информационном радиаторе», о «тихой гавани», о заказчике, то это Agile

SCRUM



ОСНОВНЫЕ КОМПОНЕНТЫ SCRUM-ПРОЦЕССА



- Роли
- Документы
- Встречи

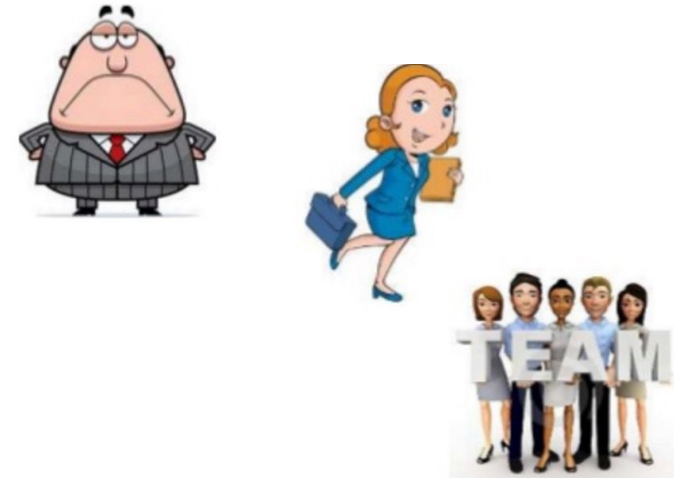
РОЛИ



Свинья идет по дороге. Курица смотрит на нее и говорит: «А давай откроем ресторан!» Свинья смотрит на курицу и отвечает: «Хорошая идея, и как ты хочешь его назвать?» Курица думает и говорит: «Почему бы не назвать „Яичница с беконом“?». «Так не пойдет, — отвечает свинья, — ведь тогда мне придется полностью посвятить себя проекту, а ты будешь вовлечена только частично».

- «СВИНЬИ»

- Скрам мастер (Scrum Master)
- Владелец продукта (Product Owner)
- Скрам-команда (Scrum Team)



- «КУРЫ»

- Пользователи (Users)
- Заинтересованные стороны (Stakeholders)
- Руководители (Managers)
- Эксперты (Consulting Experts)



SCRUM MASTER



- Отвечает за соблюдение практик и процессов
- Проводит Daily Scrum
- Ведет Burndown Chart
- Является модератором собраний
- Организует коммуникации и свободный обмен информацией

- ScrumMaster – это «владелец процесса»



*Кен Бланшар в своей книге “Лидерство к вершинам успеха” ввел термин “лидер-слуга”. Имеется ввиду, что Скрам-Мастер не привилегированный носитель власти, как это происходит в случае с классическим менеджментом, а лидер, который вовлекает участников Скрам-команды в процесс работы, не имея формальной власти.

PRODUCT OWNER (ВЛАДЕЛЕЦ ПРОДУКТА)



- Формирует видение продукта и принимает решения по продукту
- Определяет приоритеты пользовательских историй
- Разрешает спорные ситуации
- Отвечает за приемку продукта а конце каждой итерации
- Активный член команды, участвует в планировании итераций, ежедневных скрамах и ретроспективах
- Предпочтительно, если находится в одном помещении с командой



Для успешного выполнения обязанностей Владелецем Продукта все члены организации должны уважать его решения. Отражением его решений служит содержание и порядок элементов Бэклога Продукта.

РОЛЬ РУКОВОДИТЕЛЯ ПРОЕКТА



- решение вопросов размещения проектной команды
- обучение и наставничество проектной команды
- наполнение бэклога продукта вместе с РО
- создание бэклога спринта с проектной командой
- оценка стоимости, бюджетирование и контроль расходования средств, выделенных на проект
- планирование коммуникаций
- управление рисками
- управление закупками

ТИПОВЫЕ ПРОБЛЕМЫ С ЗАКАЗЧИКОМ



Владельца продукта
невозможно вовлечь в
активное участие в
проекте

Заказчик требует
пожертвовать качеством



СКРАМ-КОМАНДА (SCRUM TEAM)



Скрам-команда состоит из Владельца Продукта, Команды Разработки и Скрам-мастера.

- ✓ Скрам-команды являются самоорганизующимися и кроссфункциональными.
 - Самоорганизующиеся команды самостоятельно решают, как выполнять свою работу – без внешних указаний.
 - Компетенций кроссфункциональной команды достаточно для выполнения полного объема работ, в т.ч. весь комплекс работ в сочетании с оптимальным способом ее исполнения.
- ✓ Исполнение всех работ предполагает предварительную договоренность команды и представителей бизнеса.
- ✓ Очередная версия готового продукта должна быть доступна в любой момент времени.

ИДЕАЛЬНАЯ СКРАМ-КОМАНДА



имеет такие характеристики:

- ✓ **Самоорганизация.** Команда самостоятельно решает, как превратить Бэклог Продукта в Инкремент Продукта – без каких-либо внешних указаний (даже от Скрам-мастера).
- ✓ **Кроссфункциональность.** При этом отдельные члены команды могут обладать различными специализированными навыками и экспертизой.
- ✓ Если мы говорим об IT. "Scrum recognizes no **titles for Development Team members, regardless of the work being performed by the person.**" **VS** " **Разработчик – единственная роль** в Команде Разработки, невзирая на тип выполняемых задач." **Правило без исключений.**
- ✓ Отсутствие подкоманд в Команде Разработки. **Правило без исключений.**
- ✓ Коллективная ответственность Команды Разработки за создание Инкремента Продукта наивысшего качества.

РАЗМЕР КОМАНДЫ РАЗРАБОТКИ



Не более **9**

При подсчете численности Команды Разработки, Владелец Продукта и Скрам-мастер не учитываются, если они сами не выполняют работу из Бэклога Спринта.

ВСТРЕЧИ



СОБЫТИЯ СКРАМА (SCRUM EVENTS)



- Спринт (The Sprint)
- Планирование Спринта (Sprint Planning)
- Ежедневный Скрам (Daily Scrum)
- Обзор Спринта (Sprint Review)
- Ретроспектива Спринта (Sprint Retrospective)

СПРИНТ (THE SPRINT)



Не более **1** месяца

Спринт состоит из

- Планирования Спринта (Sprint Backlog определяется в начале итерации и не меняется до конца итерации),
- Ежедневного Скрама,
- Разработки (нет историй, выполненных на 10, 50, 95%. История или выполнена или нет),
- Обзора Спринта (в конце спринта – рабочий продукт),
- Ретроспективы Спринта.

ПЛАНИРОВАНИЕ СПРИНТА (SPRINT PLANNING)



8

Не более часов

По результатам Планирования Спринта Скрам-команда получает ответы на вопросы:

- Цель Спринта* (+ Definition of "Done")
- Каким будет Инкремент в конце Спринта?
- Как организовать работу, чтобы получить готовый Инкремент Продукта?
- Sprint backlog
- Дата демонстрации
- Место и время проведения ежедневных встреч

*Цель Спринта обеспечивает Команде Разработки достаточную гибкость касательно объема функциональности, разрабатываемой в рамках Спринта.

ЕЖЕДНЕВНЫЙ СКРАМ (DAILY SCRUM)



Не более **15** минут

Каждый участник Команды Разработки отвечает на следующие вопросы:

- Что я сделал с момента прошлой встречи для того, **чтобы помочь Команде Разработки достичь Цели Спринта?**
- Что я собираюсь сделать сегодня для того, **чтобы помочь Команде Разработки достичь Цели Спринта?**
- Какие препятствия **замедляют достижение Цели Спринта** – для меня и Команды Разработки?

ОБЗОР СПРИНТА (SPRINT REVIEW)



Не более **4** часов

Ключевые элементы:

- Участники: Скрам-команда и заинтересованные лица.
- Владелец Продукта объясняет, что готово и что нет.
- Команда Разработки обсуждает то, что получилось во время Спринта, какие были проблемы и как эти проблемы были решены.
- Команда Разработки демонстрирует готовую работу и отвечает на вопросы касательно Инкремента.
- Владелец Продукта обсуждает Бэклог Продукта.
- **Все присутствующие** обсуждают то, над чем стоит работать дальше.
- **Все присутствующие** обсуждают изменения на рынке и их потенциальное влияние на продукт. Определяются следующие действия.

РЕТРОСПЕКТИВА СПРИНТА (SPRINT RETROSPECTIVE)



Не более **3** часов

Цели Ретроспективы Спринта:

- Инспекция степени успешности Спринта относительно людей, взаимоотношений между ними, процессов и инструментов.
- Обнаружение и упорядочение того, что прошло хорошо и того, что нуждается в улучшении.
- Создание плана внедрения улучшений в процесс работы Скрам-команды.

ДОКУМЕНТЫ





Критерии готовности (Definition of Done)

Список пользовательских историй продукта (Product Backlog)

Список пользовательских историй итерации (Sprint Backlog)

Обратный график реализации историй (Burndown Chart)

DEFINITION OF DONE



Definition of Done («Критерии Готовности», DoD) — это когда все условия или Acceptance Criteria, которым должен соответствовать продукт, выполнены («Done») и готовы к принятию пользователем, клиентом, командой или потребляющей системой.

Не путаем с критериями приемки. В них требования относятся к конкретным элементам продукта со стороны клиента, а в DoD они относятся к продукту или его инкременту со стороны команды.

PRODUCT BACKLOG



Будет
реализовано

Описание требований к продукту, сформулированное в виде одного или нескольких предложений на обычном языке

Не будет
реализовано

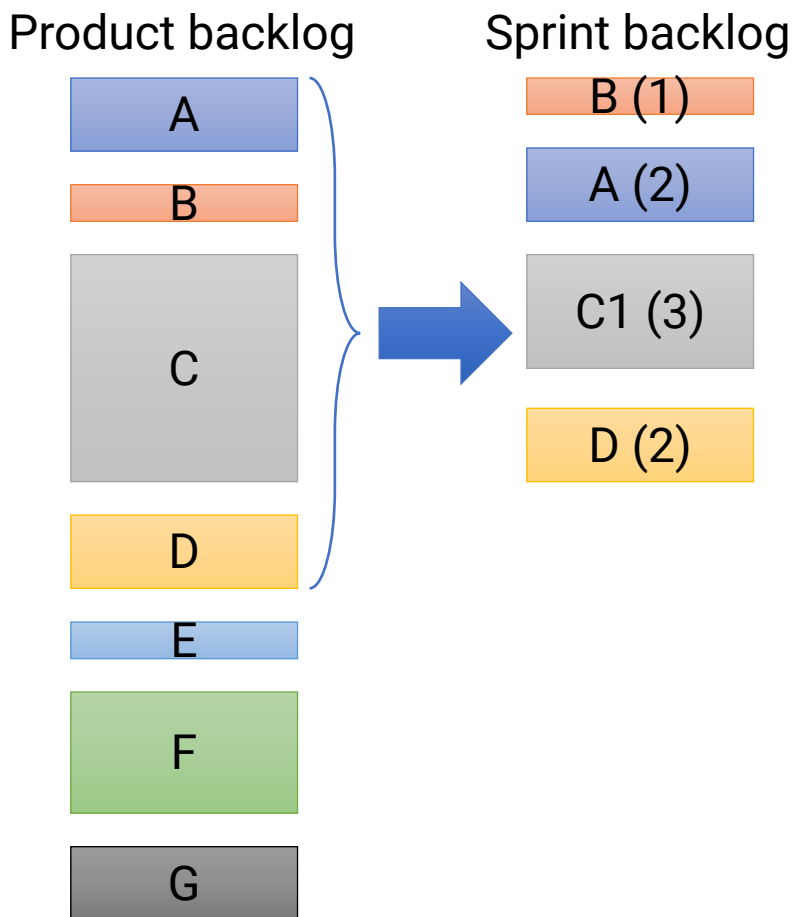
Аналог ТЗ или иерархической структуры работ верхнего уровня

Пишутся либо одобряются владельцем продукта (Product Owner)

Не решено

У каждого проекта один Product Backlog и один Product Owner

SPRINT BACKLOG



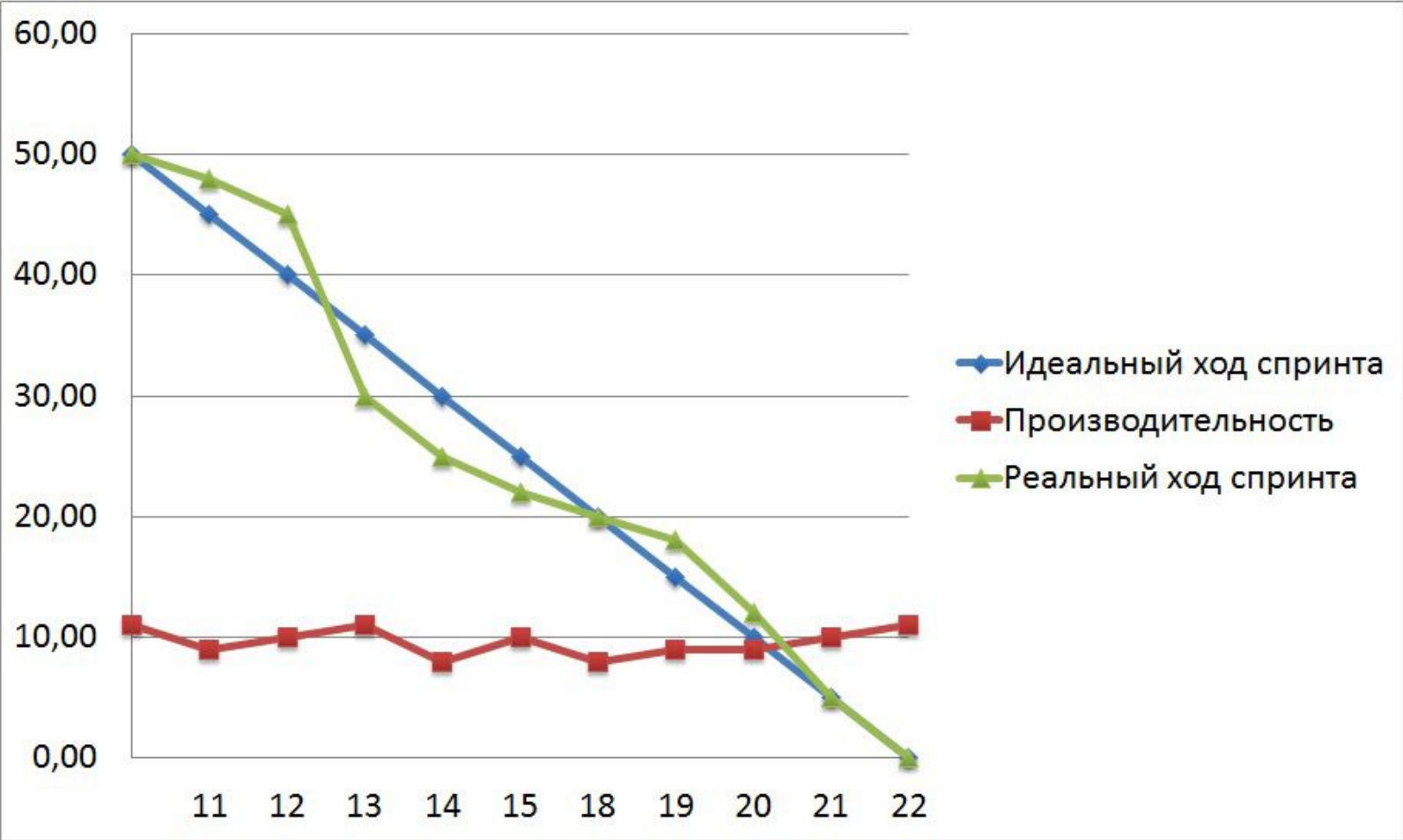
Часть Product Backlog, которая будет реализована в рамках текущего спринта

Более точные формулировки, возможно разбиение на подзадачи достаточной детализации

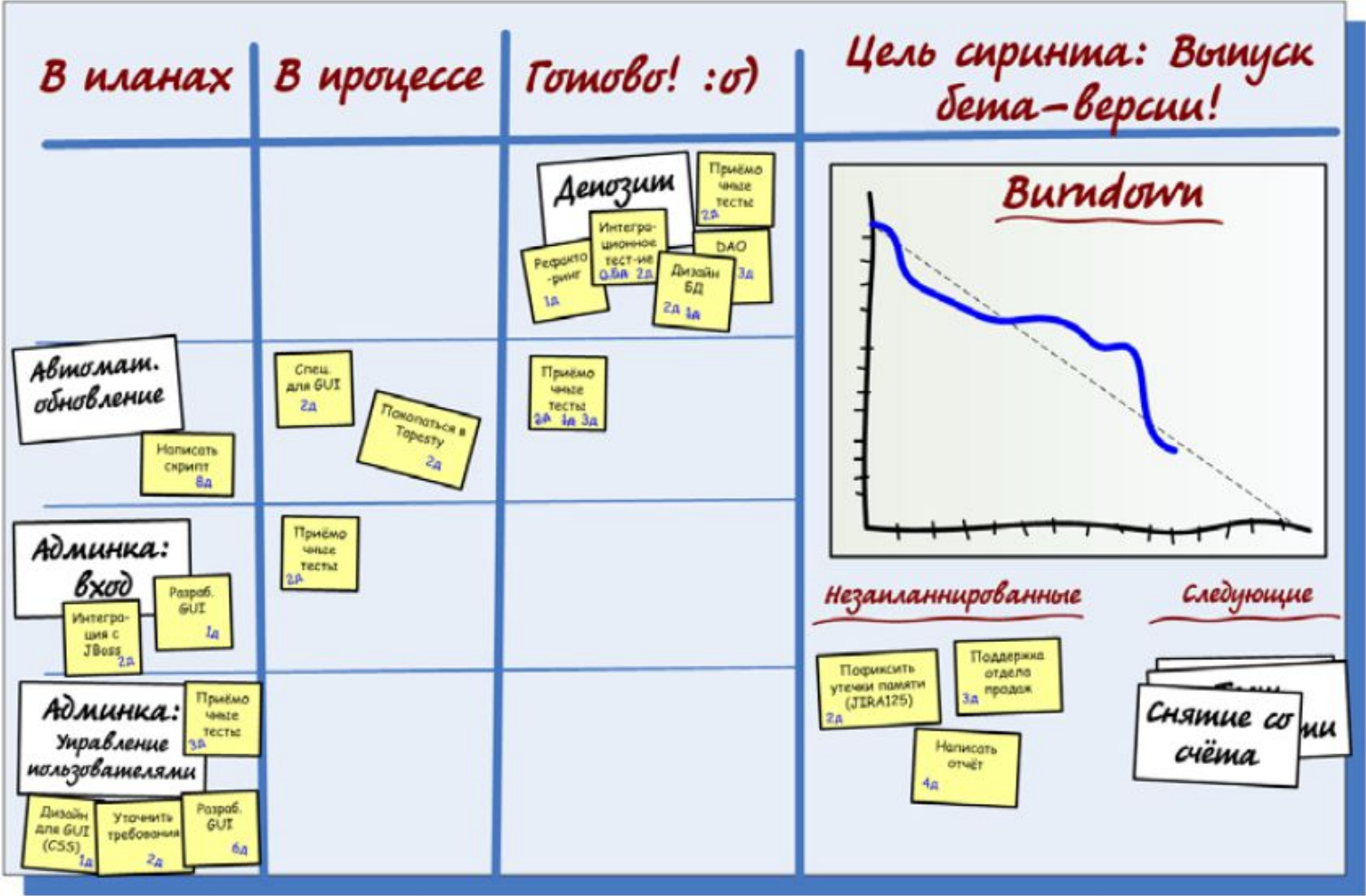
Приоритезация

Оценка трудоемкости

ОБРАТНЫЙ ГРАФИК РЕАЛИЗАЦИИ ИСТОРИЙ (BURNDOWN CHART)



СКРАМ ДОСКА (SCRUM BOARD)



LEAN ПОДХОД НА ПРИМЕРЕ КАНБАН



ОСНОВНЫЕ ПРИНЦИПЫ LEAN



- уважение к людям;
- создание ценности;
- постоянное совершенствование, так как потери похожи на гравитацию;
- 90% того, что мы делаем – это потери;
- работать спокойно, а не быстро.

НА ЧТО ЖАЛУЮТСЯ В ОПЕРАЦИОННЫХ ПРОЦЕССАХ?



- Сложно определить, где находятся «бутылочные горлышки» всех работ
- Много жалоб на качество поставляемого продукта
- Задачи неравномерно распределены во времени
- Частое «выгорание» людей: большая нагрузка, а результатов работы не видно

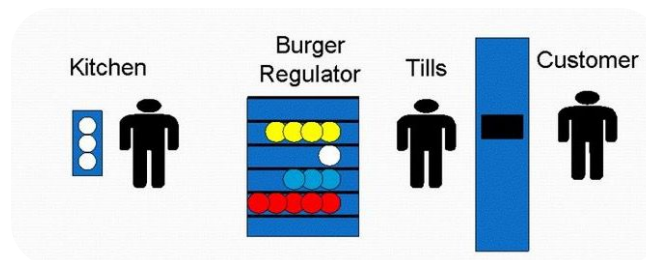
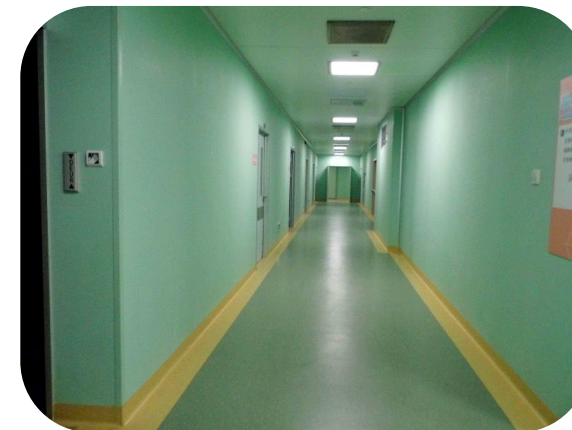
ЧТО ЭТО?



- Это не столько методология, сколько философия
- Делать только то, что действительно нужно
- Уменьшение выполняющейся в данный момент работы
- Визуализация
- Оптимизация

ВИДЫ КАНБАНА

- Канбан, как сигнал
- Канбан-система, как система ограничений
- Производственный канбан, как система борьбы с потерями
- Канбан-метод, как метод улучшения сервиса



ПОТЕРИ ПО Т.ОНО В ПРОИЗВОДСТВЕННОМ КАНБАНЕ



- muda – потери, действия, потребляющие время, усилия, ресурсы, пространство, но не создающие ценности для потребителей;
- muri – перегрузка членов команды, рабочих, сотрудников, станков и прочих мощностей при работе с повышенной интенсивностью;
- mura – неравномерность, неритмичность операций;
- дополнительный вид muda – это нерациональное использование талантов, имеющих в организации

ПРЯМЫЕ ВИДЫ ПОТЕРЬ



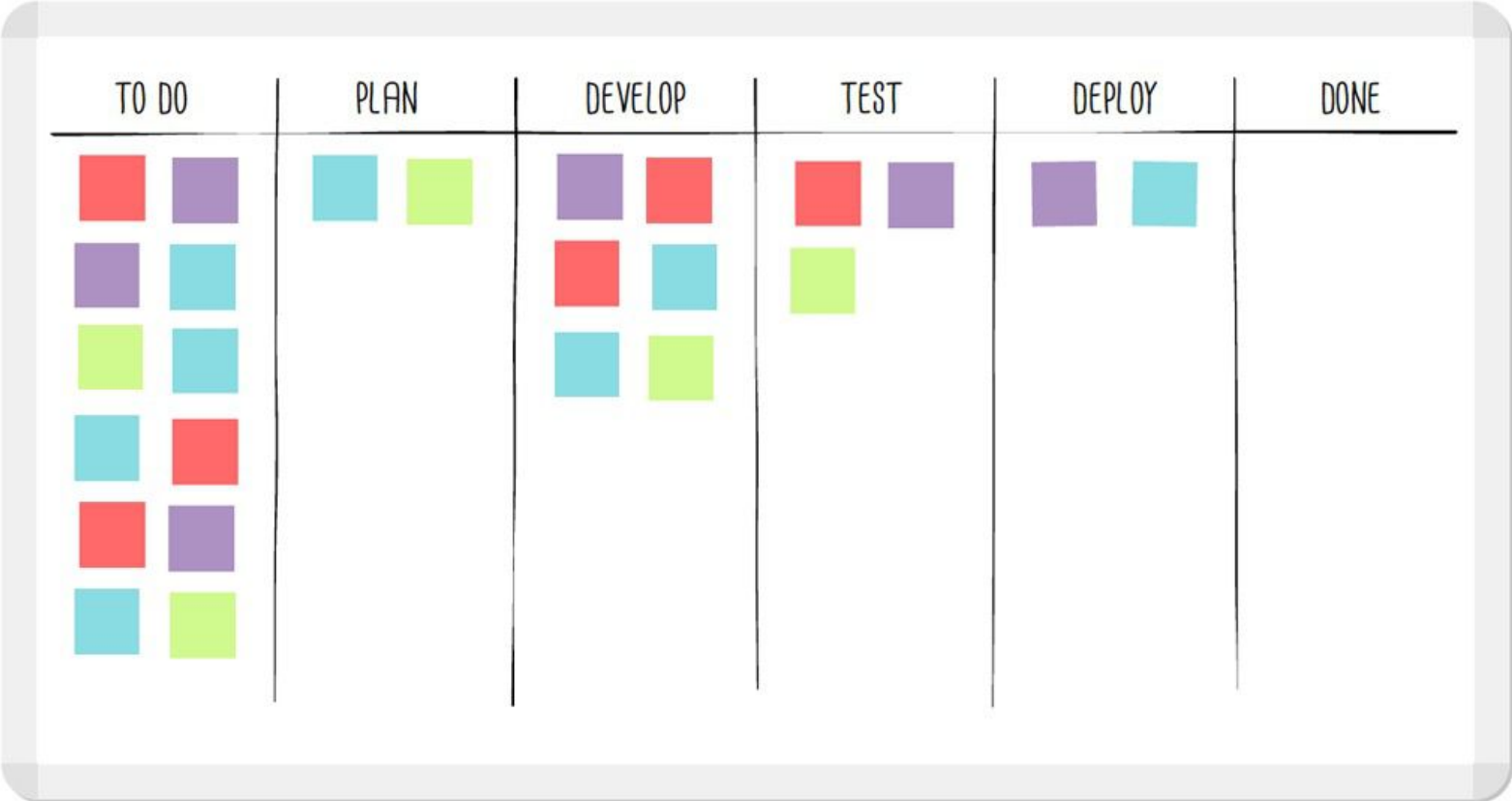
- потери из-за перепроизводства
- потери времени из-за ожидания
- потери при ненужной транспортировке
- потери из-за лишних этапов обработки
- потери из-за лишних запасов
- потери из-за ненужных перемещений
- потери из-за выпуска дефектной продукции

ОБЩИЕ ИДЕИ



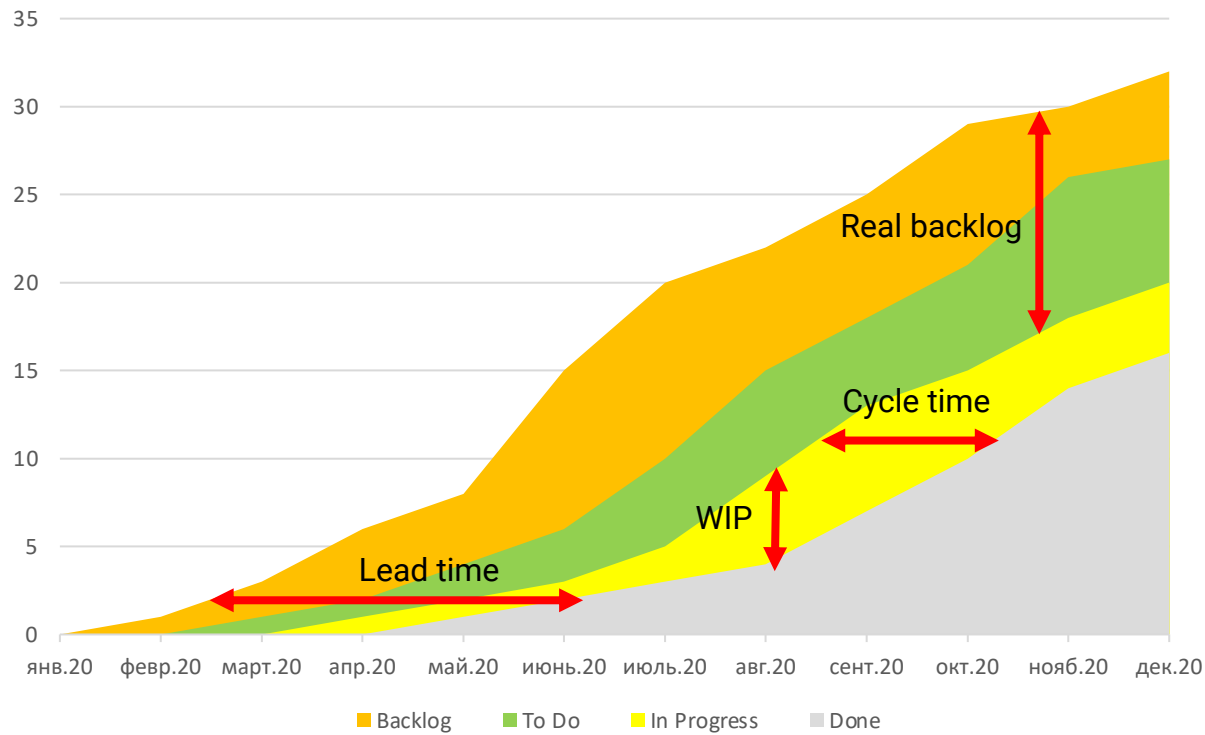
- Ограничение work-in-progress (задач, над которыми идет работа) помогает увидеть, где в системе сосредоточены проблемы
- Регулярные каденции
- Большая доска с карточками-задачами
- Доска всегда представляет актуальный «снимок» текущего статуса проекта
- Пытаемся ограничить объем работы на каждой стадии процесса
- «Бутылочные горлышки» легко идентифицировать «на глаз» благодаря скопившимся карточкам
- Все принципы нацелены на минимизацию простоя и убытков

КАК ОБЫЧНО ВЫГЛЯДИТ ДОСКА



- User Story
- Defect
- Task
- Feature

КУМУЛЯТИВНАЯ ДИАГРАММА ПОТОКА (CUMULATIVE FLOW DIAGRAM)



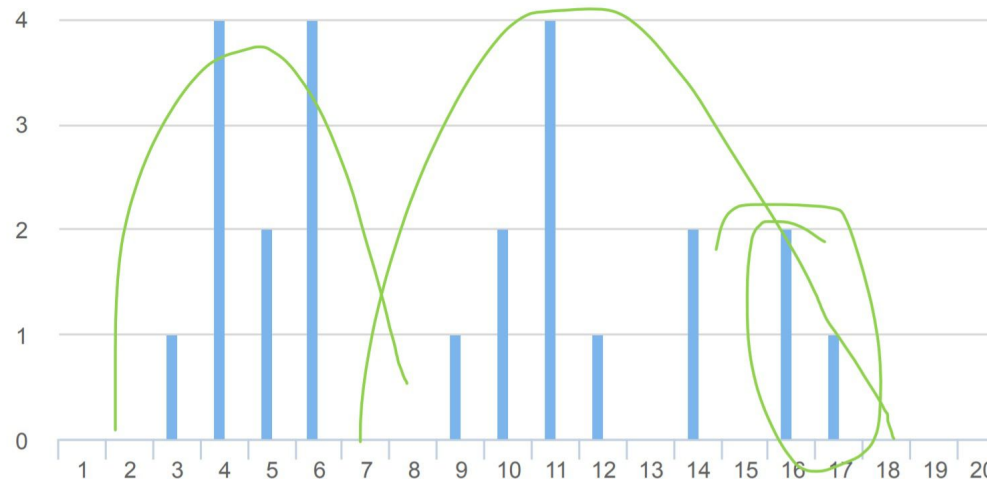
СПЕКТРАЛЬНАЯ ДИАГРАММА



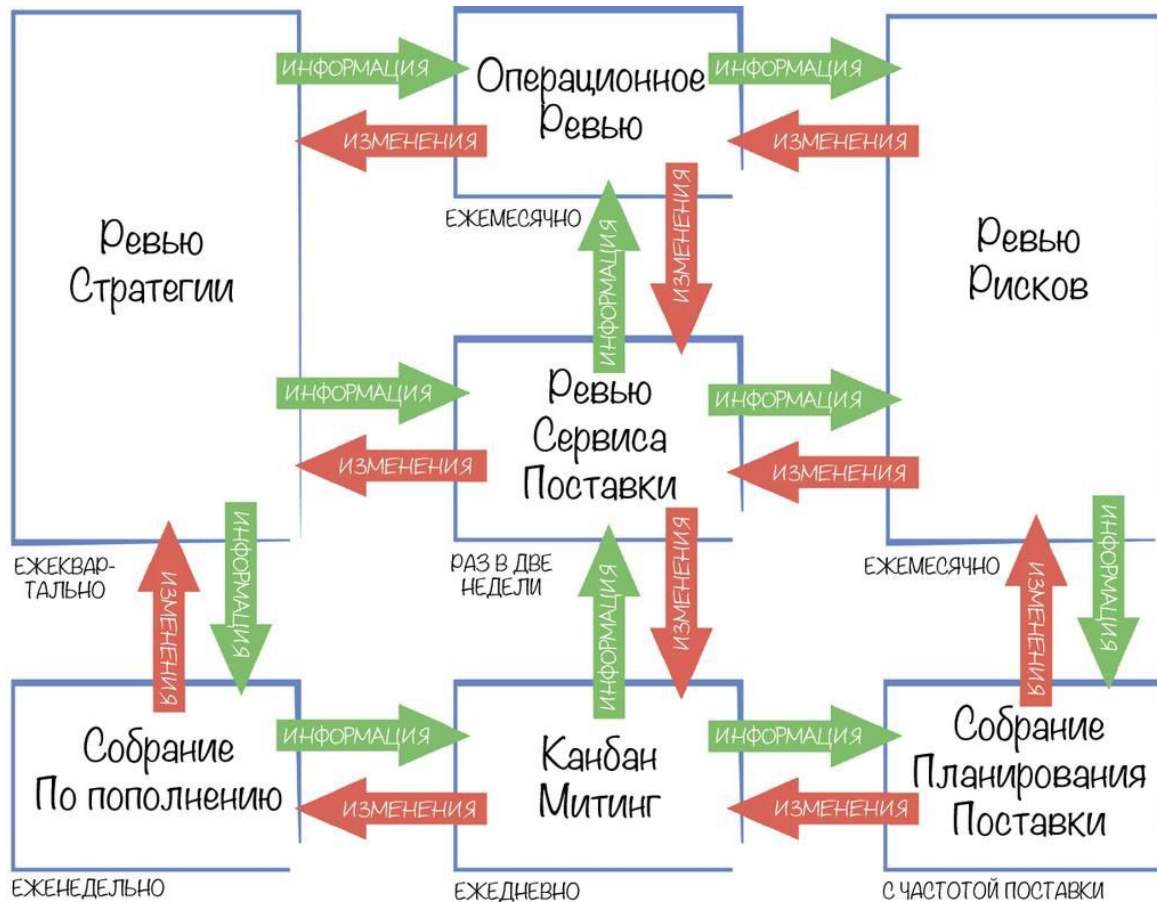
Группы куполов (кластеры задач) – это могут быть задачи разных типов, либо что-то старое, что никак не могло покинуть канбан систему, или новое

Вертикальные черточки слева – это очень простые задачи-однодневки, которые являются первыми кандидатами на автоматизацию, так как они очень простые и с ними понятно, что надо делать (если человек не включает мозг при их исполнении, то его не нужно в этих задачах вообще задействовать)

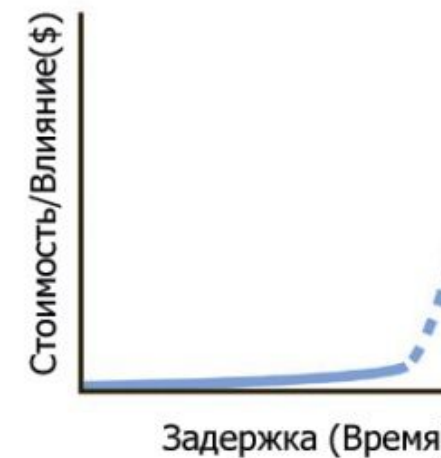
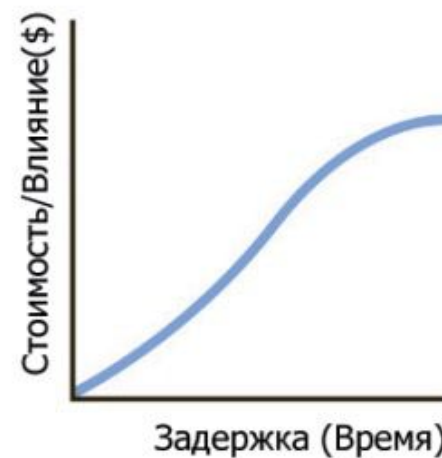
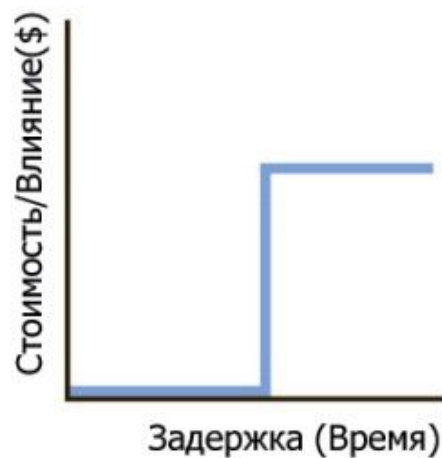
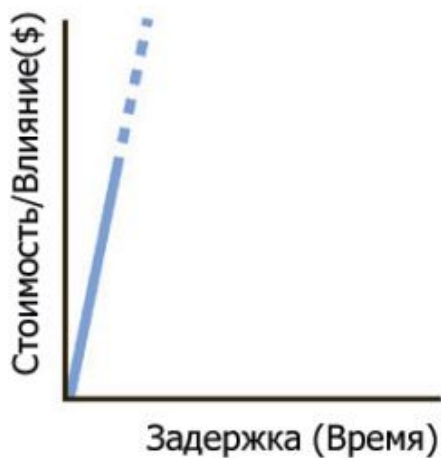
Так же можем предсказывать примерное время поставки. А вот если «забор» из одинаковых столбиков во все дни, то система в неуправляемом состоянии и время исполнения задачи может занять любое время.



ВСТРЕЧИ



СТОИМОСТЬ ЗАДЕРЖКИ



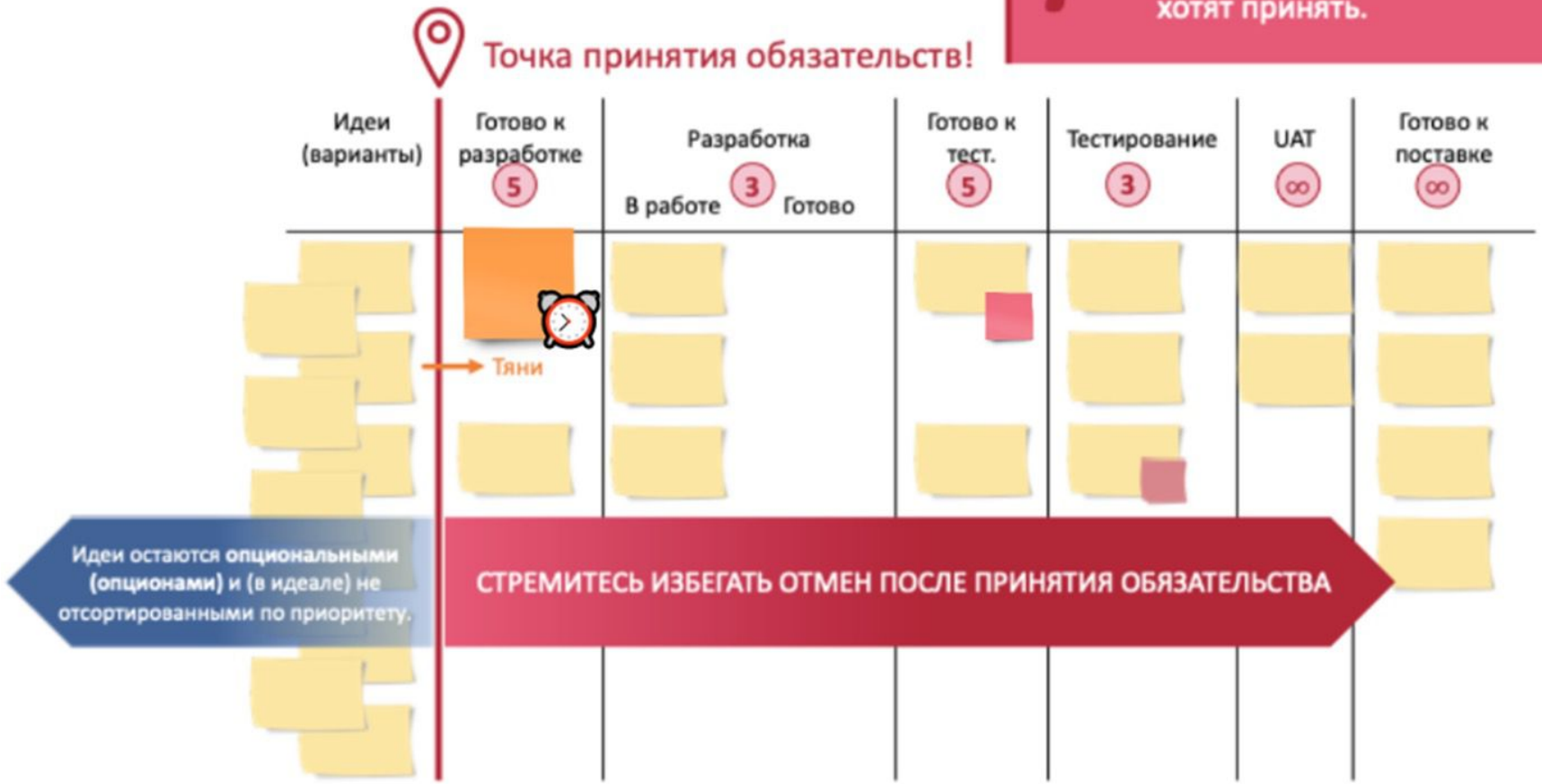
ТОЧКА ПРИНЯТИЯ ОБЯЗАТЕЛЬСТВ



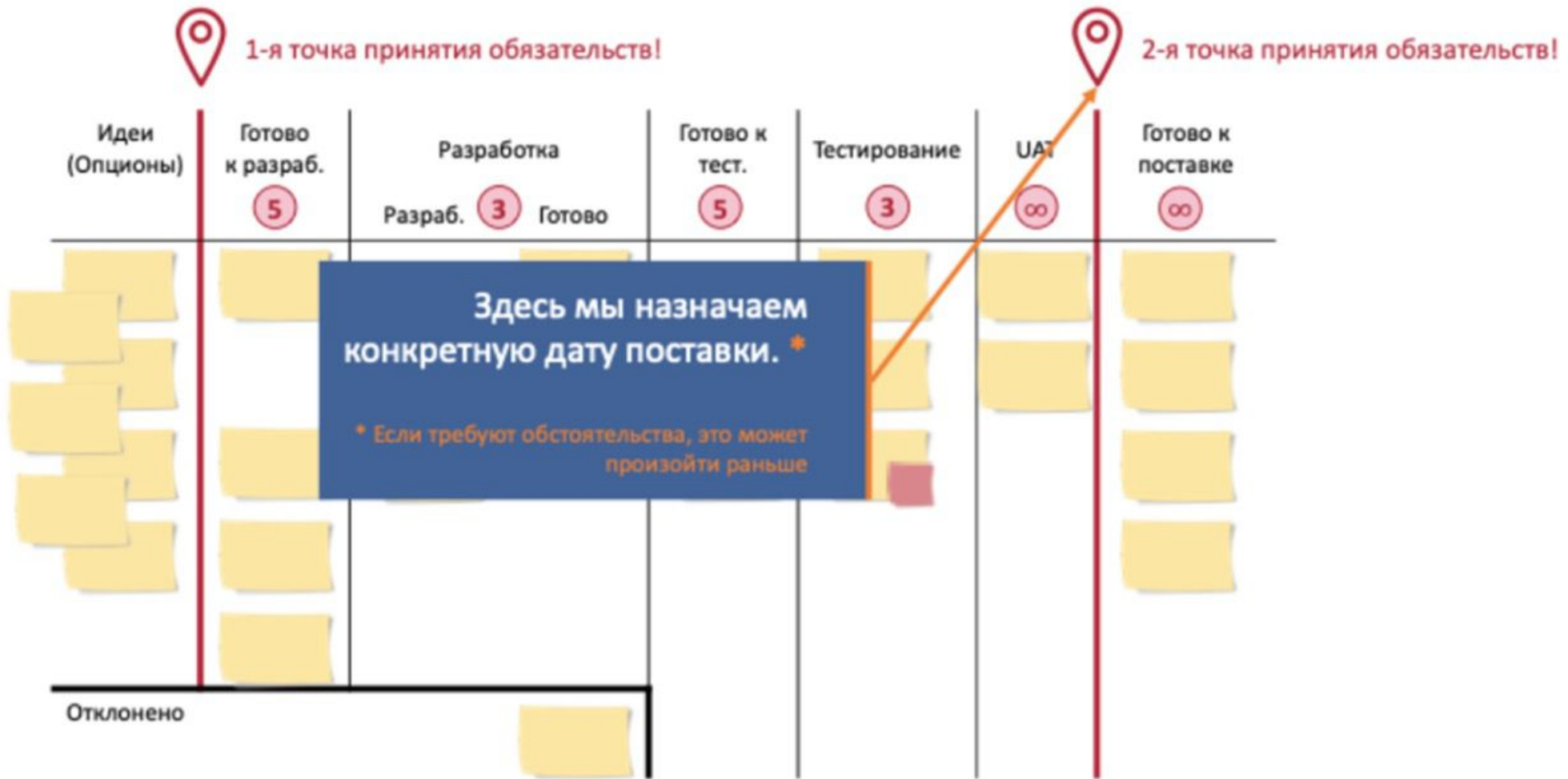
ПЕРВАЯ ТОЧКА ПРИНЯТИЯ ОБЯЗАТЕЛЬСТВ



! Мы принимаем на себя обязательство начать работу. Мы уверены, что доставку хотят принять.



ВТОРАЯ ТОЧКА ПРИНЯТИЯ ОБЯЗАТЕЛЬСТВ



МЕТОД STATIK (SYSTEMS THINKING APPROACH TO INTRODUCING KANBAN)



Сервис:

Менеджер:

Дата:

<p>1 – Источники неудовлетворённости</p> <table border="0"> <tr> <td style="vertical-align: top;"> <p>Внутренние Что вас не удовлетворяет внутри сервиса?</p> </td> <td style="vertical-align: top;"> <p>Внешние Кто неудовлетворен вашей работой вовне? С кем у вас возникают конфликты?</p> </td> </tr> </table>						<p>Внутренние Что вас не удовлетворяет внутри сервиса?</p>	<p>Внешние Кто неудовлетворен вашей работой вовне? С кем у вас возникают конфликты?</p>	<p>3 – Рабочий поток</p> <p>Нарисуйте рабочий процесс для каждого типа рабочего элемента. Есть ли сходства и различия между ними? Есть ли одновременная и неупорядоченная деятельность? Есть ли внешние зависимости, риски и т. д.?</p>						<p>4 – Классы обслуживания</p> <p>Для каждого типа рабочего элемента укажите текущие классы обслуживания, их политики и ожидания по поставке.</p>																																							
<p>Внутренние Что вас не удовлетворяет внутри сервиса?</p>	<p>Внешние Кто неудовлетворен вашей работой вовне? С кем у вас возникают конфликты?</p>																																																				
<p>2 – Анализ запросов</p> <table border="1"> <thead> <tr> <th>Тип рабочего элемента</th> <th>Источник</th> <th>Пункт назначения</th> <th>Частота поступления</th> <th>Природа запроса</th> <th>Ожидания клиента</th> </tr> </thead> <tbody> <tr><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td></tr> </tbody> </table>						Тип рабочего элемента	Источник	Пункт назначения	Частота поступления	Природа запроса	Ожидания клиента																																					<p>5 – Каденции по пополнению и поставке</p> <p>Как часто вы сейчас пополняете очередь? Как часто поставляете? Укажите частоту для каждого типа рабочего элемента.</p>					
Тип рабочего элемента	Источник	Пункт назначения	Частота поступления	Природа запроса	Ожидания клиента																																																
<p>6 – Визуализация Канбан-системы</p> <p>Этот раздел предназначен для простого эскиза, помогающего команде доставки, менеджеру и тренеру определить основные контуры визуальной платы. Они могут включать в себя плавательные дорожки, двухуровневую структуру, использование цвета и т. д. Нет необходимости делать этот раздел миниатюрной копией фактической доски.</p>																																																					

Сервис: Менеджер: Дата:

Срочный
Стандартный
Фикс дата

1 – Источники неудовлетворённости

Внутренние
Что вас не удовлетворяет внутри сервиса?

- Крупные задачи в разных этапах работы, выделенные ресурсы, что не актуально, нет информации о актуальности задачи, не согласование с заказчиком
- WIP-лимитов нет
- Много задач в статусе "готовы к деплою/размещению", но они стали не актуальными, а актуальные только "догоняют"

Внешние
Кто неудовлетворен вашей работой вовне? С кем у вас возникают конфликты?

- Срок поставки
- Качество
- Неудовлетворённость качества поставок (детализированных описаний задач)

3 – Рабочий поток

Нарисуйте рабочий процесс для каждого типа рабочего элемента. Есть ли сходства и различия между ними? Есть ли одновременная и неупорядоченная деятельность? Есть ли внешние зависимости, риски и т. д.?

Детализация (краткое описание из тз, общение с заказчиком)

Дизайн (визуализация)

Согласование

Разработка ТР

Разработка (кодирование)

Тестирование

Подготовка демо сценария

Демонстрация

- заказчик
- аналитик
- дизайнер
- разработчик
- тестировщик
- архитектор

4 – Классы обслуживания

Для каждого типа рабочего элемента укажите текущие классы обслуживания, их приоритет и ожидания

- Запрос фичи
- Запрос изменения
- Хотфикс, запрос изменений
- Запрос на экспедирование

5 – Каденции по пополнению и поставке

Как часто вы сейчас пополняете очередь? Как часто поставляете? Укажите частоту для каждого типа рабочего элемента.

6 – Визуализация Канбан-системы

Этот раздел предназначен для простого эскиза, помогающего команде доставки, менеджеру и тренеру определить основные контуры визуальной платы. Они могут включать в себя плавательные дорожки, двухуровневую структуру, использование цвета и т. д. Нет необходимости делать этот раздел миниатюрной копией фактической доски.

2 – Анализ запросов

Тип рабочего элемента	Источник	Пункт назначения	Частота поступления	Природа запроса	Ожидания клиента
Запрос фичи	Заказчик	Заказчик	3 в неделю	регулярно	3 недели
Запрос изменения	Заказчик	Заказчик	по запросу	рандомно	2 мес
Хотфикс, запрос изменений	Заказчик	Заказчик	по запросу (1 в мес)	рандомно	1 день
	Заказчик	Заказчик	один раз	хаотик	в указанную дату

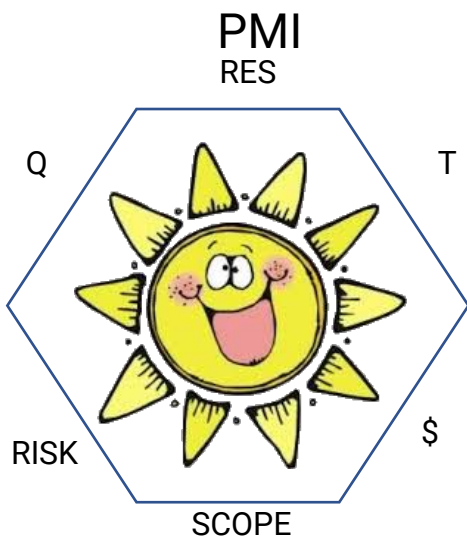


ПРАКТИКА. РАБОТА С ДОСКОЙ И ГРАФИКАМИ



СРАВНЕНИЕ ГИБКИХ И ТРАДИЦИОННЫХ ПОДХОДОВ

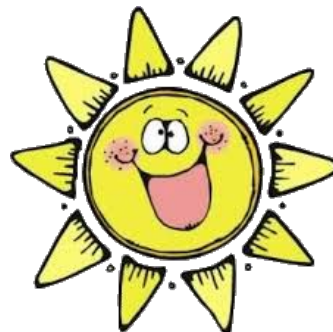
ОБЩИЙ ВЗГЛЯД



Удовлетворить стейкхолдеров
Уложиться в ограничения:

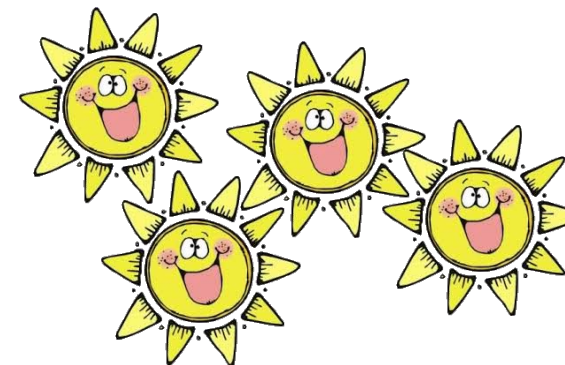
- Сроки
- Бюджет
- Скоуп
- Риски
- Качество
- Ресурсы

Scrum



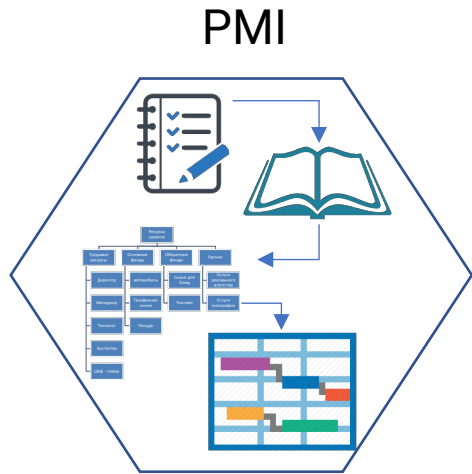
Удовлетворить стейкхолдеров

Kanban



- Выстроить процесс
- Сформировать поток
- Непрерывно совершенствовать процесс производства

РОЛИ

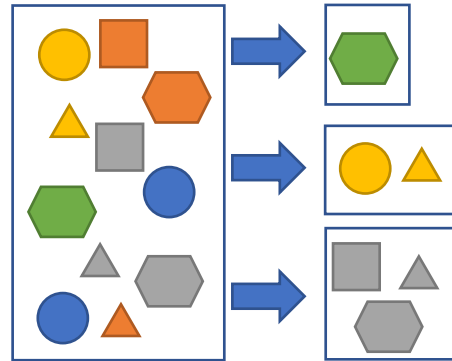


Project manager
(менеджер проекта)

Sponsor (спонсор)

Project team
(команда проекта)

Scrum

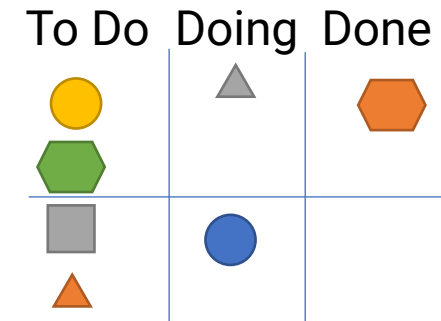


Product owner (владелец продукта)

Scrum master (скрам-мастер)

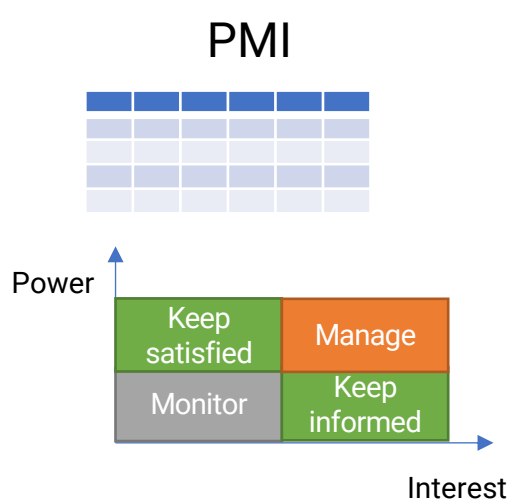
Project team (команда проекта) -
кроссфункциональная

Kanban



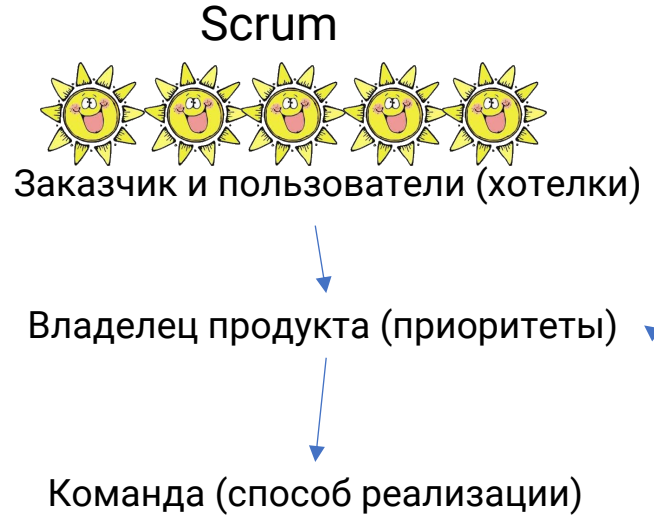
Project team (команда проекта) – разделение по фазам

ЗАИНТЕРЕСОВАННЫЕ СТОРОНЫ



Реестр и матрица заинтересованных сторон:

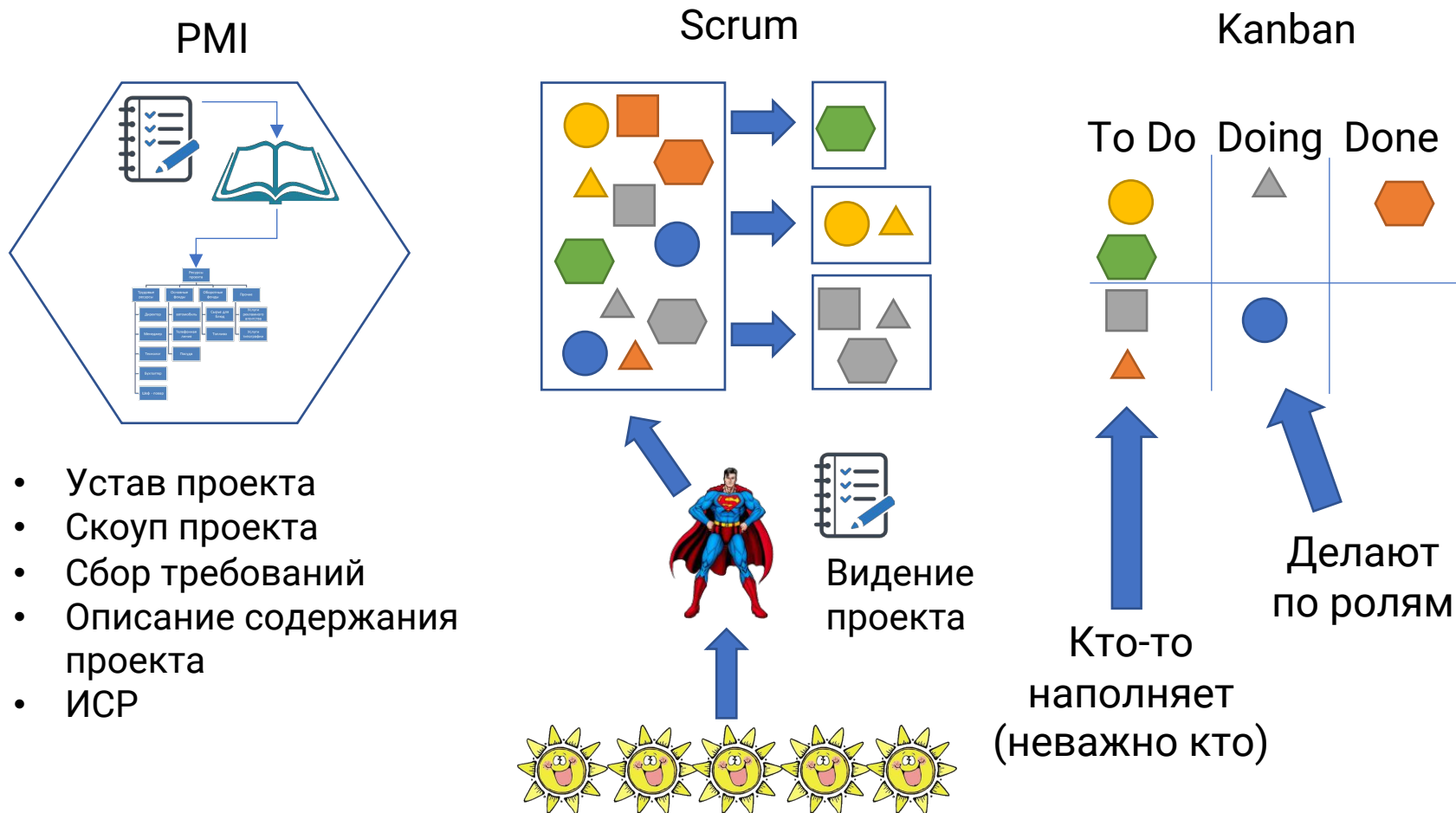
- Спонсор
- Менеджер проекта
- Заказчик и пользователи
- Команда
- Конкуренты
- ...



Kanban

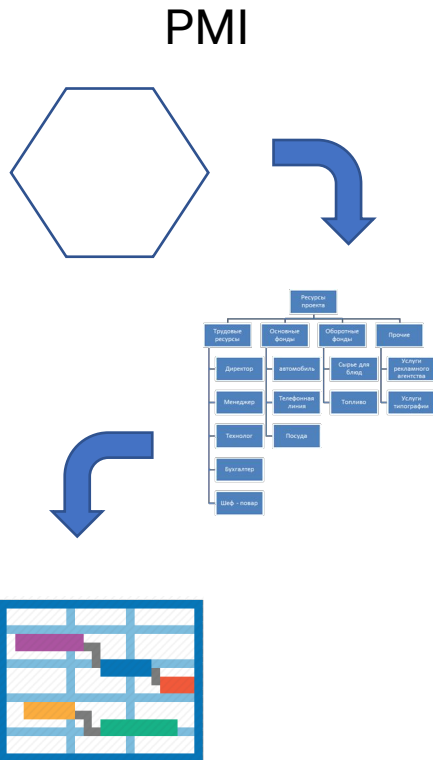
Специально не смотрим

СОДЕРЖАНИЕ

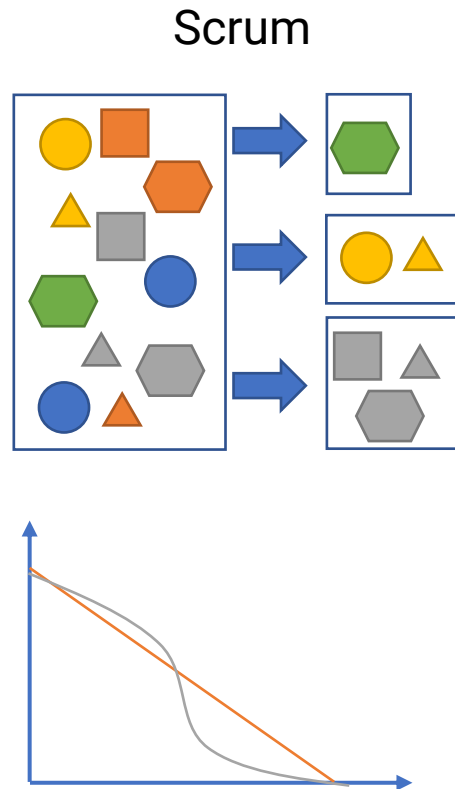


- Устав проекта
- Скоуп проекта
- Сбор требований
- Описание содержания проекта
- ИСР

СТОИМОСТЬ



Различные способы оценки стоимости
Спросить поставщиков
Резервы на риски



Себестоимость =
зарплата + налоги

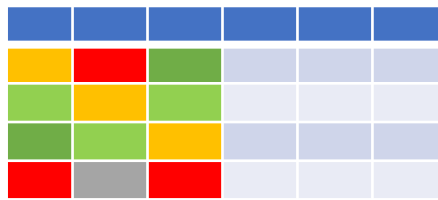
Kanban

Специально
не смотрим

РИСКИ

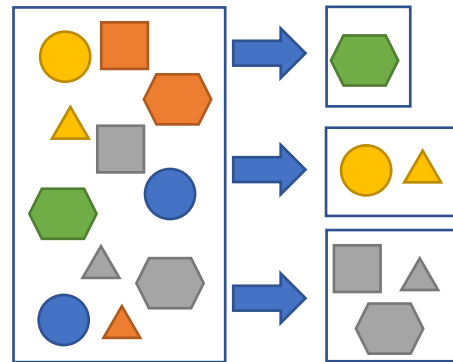


PMI



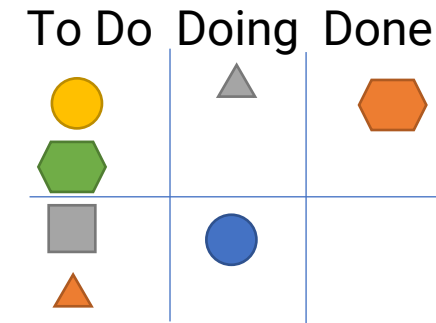
Реестр рисков и возможностей
Идентификация рисков
Качественный анализ
Количественный анализ
Планирование реагирования
Резервы
База знаний

Scrum



Специально не
смотрим, но
есть
ретроспективы

Kanban



Визуализация
узких мест

КАЧЕСТВО



PMI

7 инструментов контроля качества:

- Диаграмма Ишикавы
- Диаграмма разбрасывания
- Диаграмма Парето
- Гистограмма
- Чек-лист
- Контрольная карта
- Блок-схема

QA и QC

- Хороши ли наши практики

Ответственность менеджера, вовлечена вся команда (TQM). Отдельные роли могут быть сосредоточены на проверка результатов в интересах пользователей и заказчиков

Scrum

- Проверка владельцем продукта во время демо
- Ретроспективы похожи на работу с качеством

Командная ответственность + владелец продукта как адвокат пользователя

Kanban

Настройка ролей, входов и выходов

Команда стремится к тотальному обеспечению качества (TQM), менеджер помогает + отдельные роли могут фокусироваться именно на проверке результатов в интересах пользователей и заказчиков

ЗАКУПКИ



PMI

- Планирование закупки
- Осуществление закупки
- Контроль закупки
- Закрытие закупки

Типы контрактов

- Fixed Price
- T&M
- CR

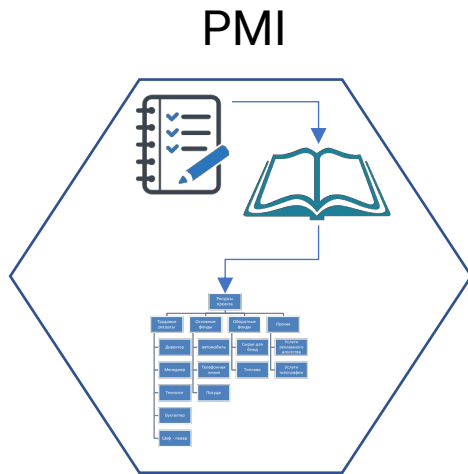
Scrum

- Нужно договориться с внешней командой договориться работать по scrum
- Некоторые закупки не равны вовлечению внешней команды, а требуют долгого и тщательного планирования

Kanban

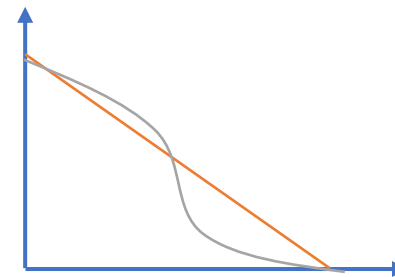
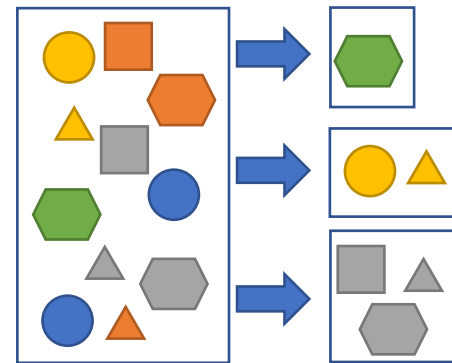
Не рассматривается

ПРОГРЕСС



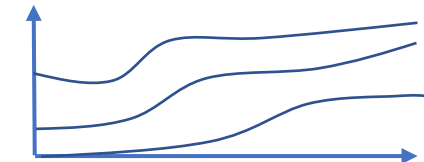
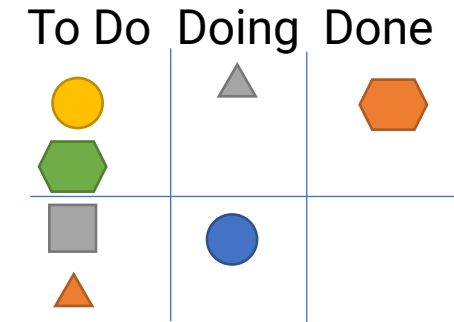
- Различные способы отслеживания планов
- Отчеты по запросы
 - Правила заполнения
 - Метод освоенного объема EVA
 - SPI, SV, CPI, CV

Scrum



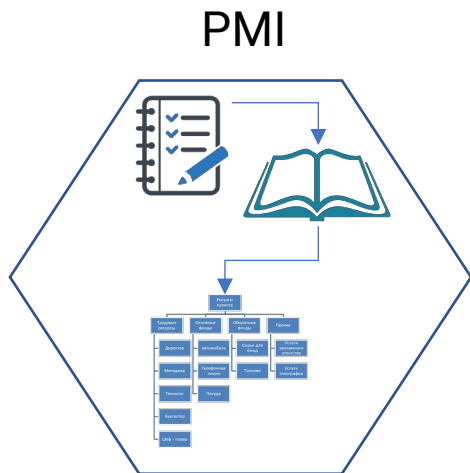
Burndown chart
Burnupchart
Agile EVA

Kanban



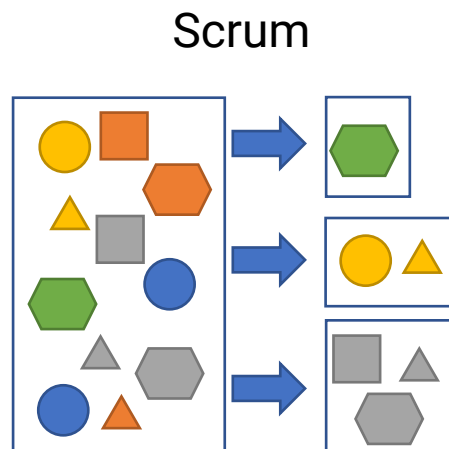
TTM (time to market)
LT (lead time)
CT (cycle time)
WIP (work in progress)
Кумулятивная
диаграмма потока

ИНТЕГРАЦИЯ

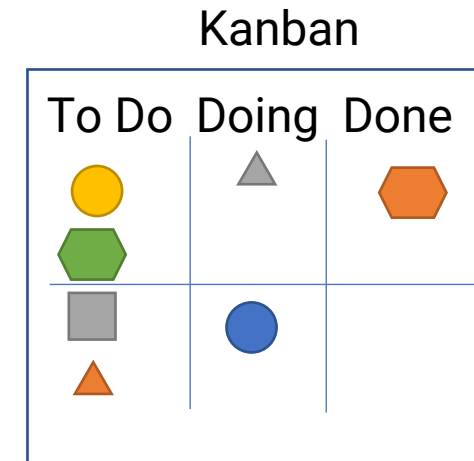


Менеджер проекта

- интегратор
- формирует и удерживает команду специалистов
- контролирует связанность планов, их взаимное влияние и соответствие ограничениям

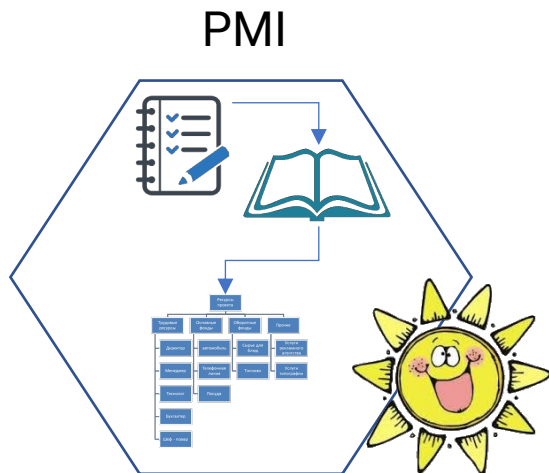


- Команда сама организует работу в итерации согласно приоритетам владельца продукта
- Владелец продукта – интегратор в рамках продукта



Тот, кто совместно с командой придумал и настроил поток задач, тот, кто обеспечивает стык с другими методами и командами, часовщик

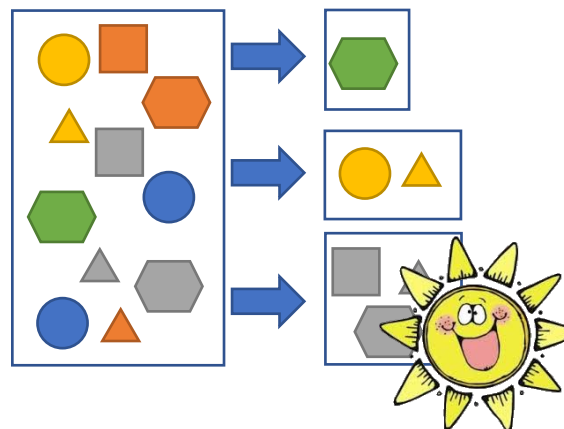
ЗАКРЫТИЕ



Сдача-приемка

- На всем протяжении проекта
- Формальное завершение проекта
- Высвобождение ресурсов и команды
- Фиксация Lessons Learned

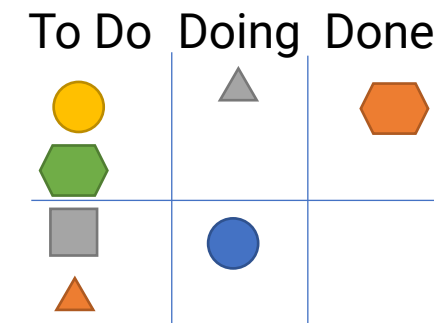
Scrum



После каждой итерации

- Демонстрация
- Ретроспективы

Kanban

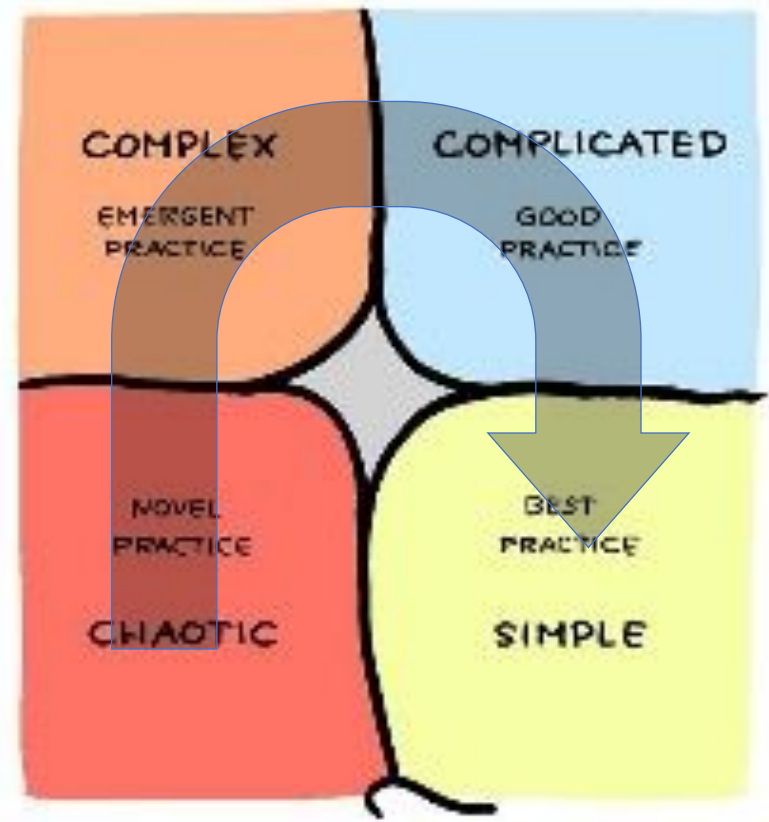


Результаты работы по приоритетам

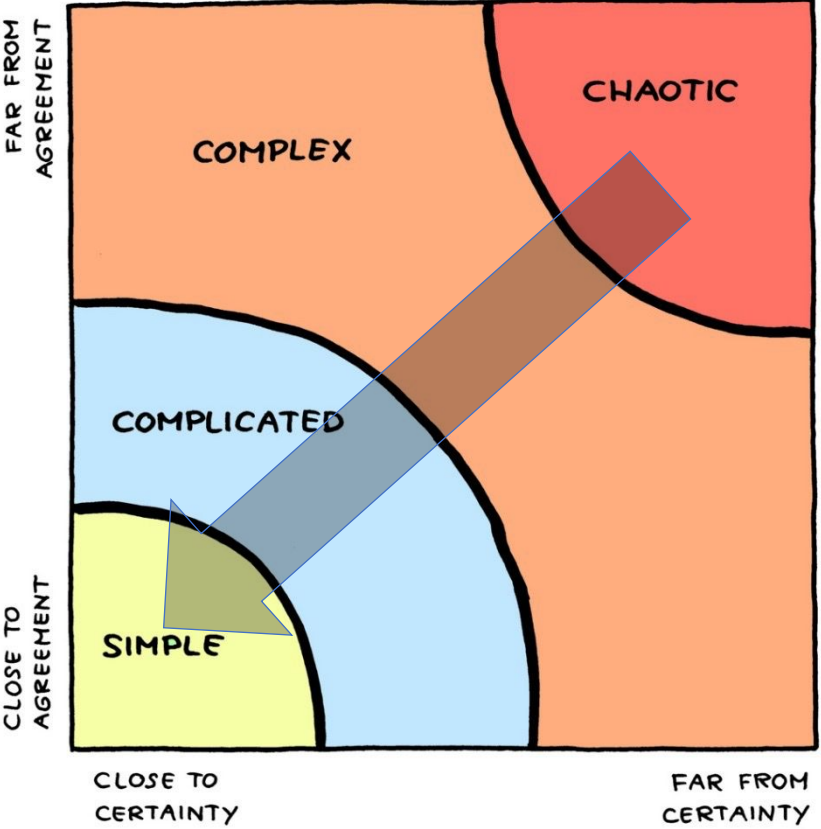
МОДЕЛЬ СЛОЖНОСТИ ПРОЕКТОВ



Cynefin framework



The Stacey complexity model



ВСЕ ЗАВИСИТ ОТ ТИПА ПРОЕКТА И ВЗАИМОДЕЙСТВИЯ С ЗАКАЗЧИКОМ, НАПРИМЕР



Характеристики проекта	Традиционный подход	Agile внутри	Agile снаружи
Ограничения по цене, времени, требованиям	++	0	-
Внешний заказчик, ограничения по времени	0	++	+
Внутренний заказчик, ограничения по времени	-	+	++


СРАВНЕНИЕ SCRUM И KANBAN



	Scrum	Kanban
Описание задач	Прорабатываются заранее	Может изменяться
Длительность задач	По возможности одинаковые	Могут быть разные
Перенос задач	Задачи должны быть сделаны в рамках итерации	Выполняются столько, сколько требуется
Расписание встреч	Набор обязательных встреч	По необходимости
Ключевая метрика	Скорость выполнения задачи	Время выполнения задачи
Члены команды	Могут быть разрозненны	Сплочённый коллектив
Вид команд	Кросс-функциональные	Кросс- функциональные или профессиональные
Взятие задач в работу	Группой в начале итерации	В любой момент времени
Роли в команде	Жестко определены	Определены условно и по обстоятельствам
Ограничения	Задачи берутся в спринт, Количество не важно	Количество одновременно выполняемой работы на этапе ограничено
Этапы	Сделать, в работе, выполнено	Определяются для проекта индивидуально
Дедлайны	Соблюдаются на уровне задач в итерации	Большая ориентация на процесс

ОЦЕНКИ





Хорошая оценка – это такая оценка, которая находится внутри $\pm 25\%$ реального срока в 75% случаев.

SEMM

ТИПИЧНАЯ СИТУАЦИЯ



- Шеф:
 - Данный модуль должен быть сделан через три месяца, к первому сентября, вот список функций. Через сколько времени модуль будет готов?
- Работник:
 - Если у меня не будет проблем с разработкой, пауз и других непредвиденных обстоятельств, то я смогу его закончить через 6 месяцев
- Шеф:
 - Но нам нужен этот модуль к первому сентября
- Работник:
 - Ok...



ЧТО НА САМОМ ДЕЛЕ



Цель – выпустить версию к 1 сентября

Оценки смешиваются с целями: «сделать через 3 месяца»

ЧТО В ИТОГЕ?



С большой долей вероятности модуль не будет сделан к сроку, как бы ни старались, просто физически не хватит времени, даже если работать круглосуточно

Как следствие – недовольный шеф

Если вдруг проявили героизм и сделали, то героем никто никого считать не будет

ЧТО ДЕЛАТЬ?



Спросить шефа о том, почему так критично 3 месяца (может быть, будет выставка и нужно демо)

По возможности адекватно оценить каждую функцию отдельно, выстроить их по приоритетам и выбрать те, которые нужны в первую очередь

Не бояться шефа и не подгонять свои оценки под его цели, иначе будет еще хуже

В ЧЕМ ЛУЧШЕ МЕРИТЬ



<https://scrumtrek.ru/userfiles/reports/AgileSurvey18.pdf>

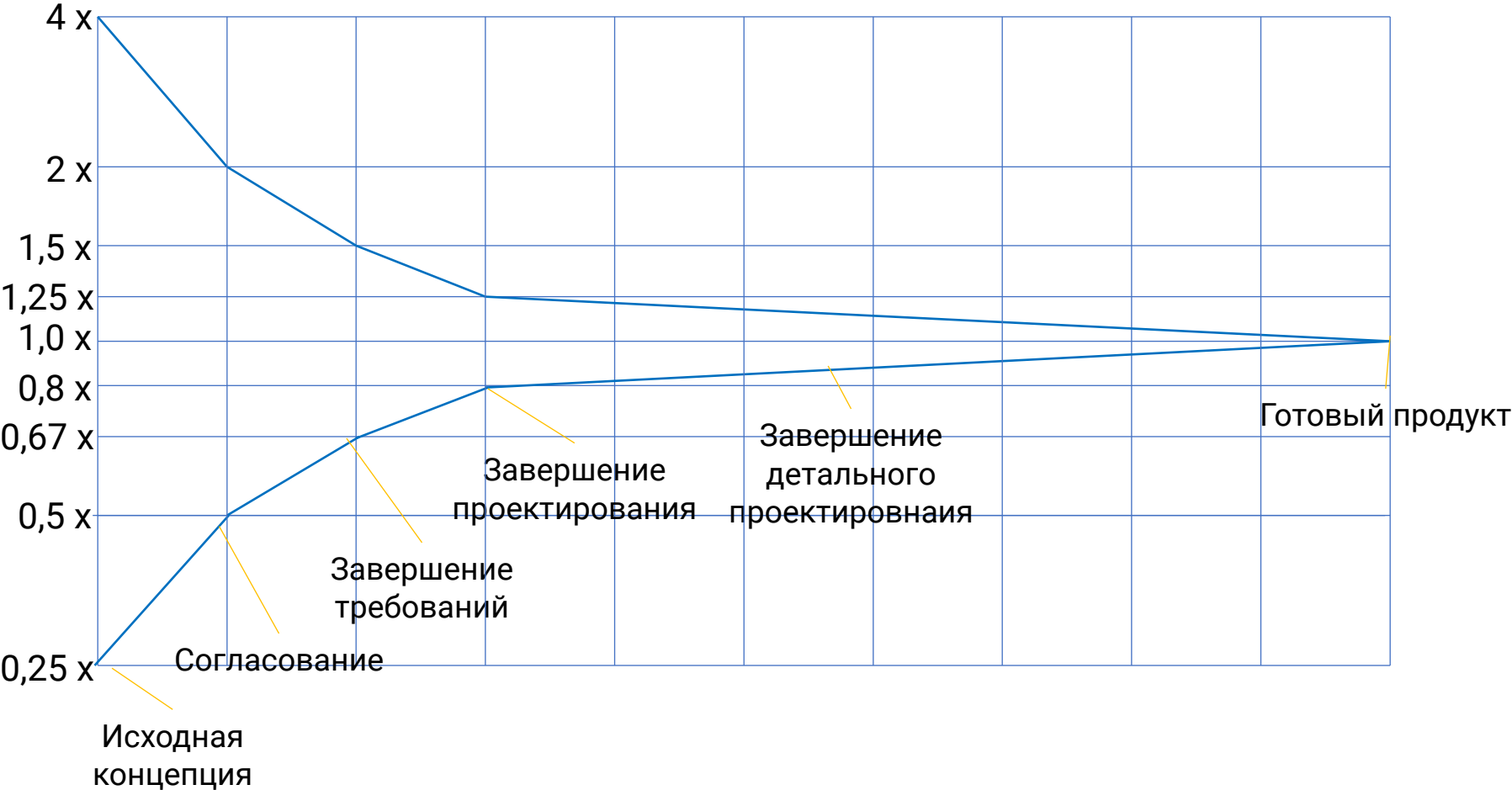
НА ЧТО ОБРАЩАТЬ ВНИМАНИЕ



Оценки – это не точные вычисления сроков, трудоемкости, объемов работ, а именно оценки. Это приблизительные значения, которые могут находиться в довольно широком диапазоне

Вычисления могут дать более точные значения, но они так же могут не учитывать множество факторов и аналогично определяются в диапазоне

ПОМНИТЕ О КОНУСЕ НЕОПРЕДЕЛЕННОСТИ



ПОРОЧНЫЙ ПРОЦЕСС ОЦЕНКИ



Задачи, которые не попали в оценку, но все равно существуют в проекте

Важные фоновые задачи игнорируются: настройка инфраструктуры, как самый частый пример

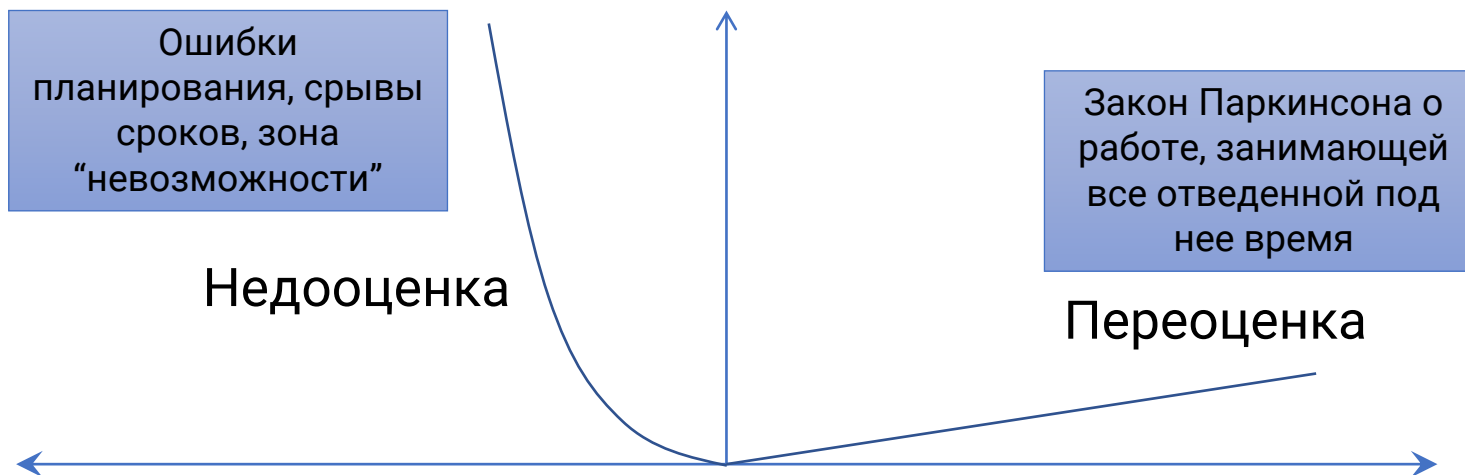
Учитывается только время на непосредственное выполнение задач, но не на коммуникацию и обсуждения

КАК НЕ ПЕРЕПУТАТЬ ОЦЕНКУ И ПЛАНИРОВАНИЕ?



- Оценка должна быть непредвзятой и аналитической
- Планирование – это агрессивный целеориентированный процесс
- Планы часто выдаются за оценки
- И даже хуже – часто планы конвертируются в расписание проекта

НЕДООЦЕНКА И ПЕРЕОЦЕНКА



ОЦЕНКИ НУЛЕВОГО ПОРЯДКА (ZERO ORDER ИЛИ RAW ESTIMATIONS)



Иногда нет возможности уделить оценке хотя бы полчаса, например, в случае, если кто-то спрашивает о том, стоит ли вообще браться за некий проект и сколько это может теоретически занять

В этом случае используются raw estimations, которые могут дать представление только о порядке величины трудоемкости (неделя, две, месяц, полгода...)

ПОДСЧЕТ, ВЫЧИСЛЕНИЕ, ЭКСПЕРТНАЯ ОЦЕНКА



Те самые задачи про то, сколько шариков от пинг-понга нужно, чтобы заполнить автобус, или за какую сумму денег можно взяться перемыть все окна в небоскребах города...

КАЛИБРОВКА



Если есть возможность отслеживаем, сколько времени мы реально потратили на ту или иную задачу, и корректируем себя

Сравниваем с историческими данными

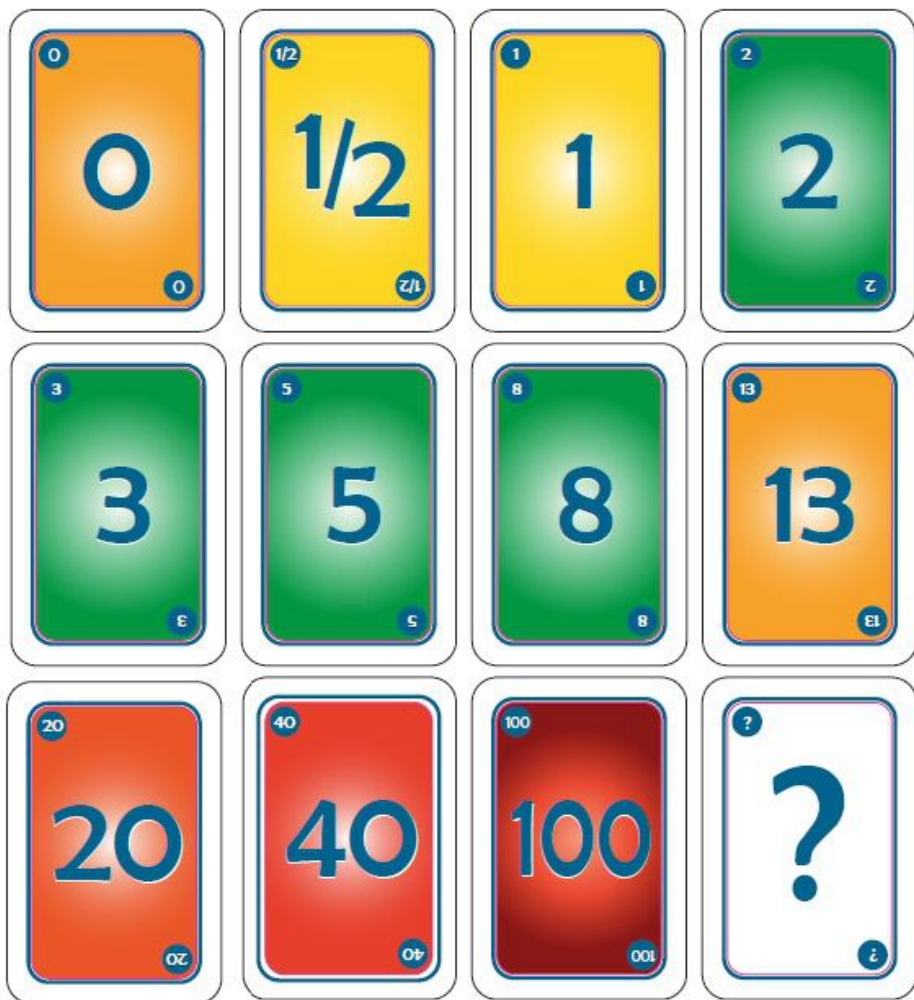
Сравниваем со средними данными по отрасли

ОЦЕНКА ПО АНАЛОГИИ



Если подобный проект уже когда-либо был реализован, то можно посмотреть на WBD и бэклоги для него, вычленив характерные части и использовать в новых оценках

PLANNING POKER



ПОШАГОВЫЙ АЛГОРИТМ



- Каждому разработчику дается набор карт с числами
- Выбирается атомарная задача, стоимостью в 1
- Ведущий описывает задачу, и команда кратко уточняет ее
- Каждый участник выбирает свою оценку и кладет карту рубашкой вверх.
 - Оценка – это размер задачи по сравнению с атомарной
 - Некоторые задачи могут быть меньше атомарной, тогда их можно объединить в пакет
- Все одновременно открывают карты
- Если есть разница (большая), то обсуждается «Почему?»
- Делаем еще один раунд оценки, пока все не сойдутся на одном значении

ПРИНЦИПЫ ОЦЕНОК В AGILE ПРОЕКТАХ



Относительные оценки

Коллективные оценки и обсуждение, planning poker

Частые переоценки (помним о конусе неопределенности)

Оценки отдельных небольших задач, а не проекта, спринта или большой пользовательской истории в целом

МЕТРИКИ



НЕЛЬЗЯ УЛУЧШИТЬ ТО, ЧТО НЕЛЬЗЯ ИЗМЕРИТЬ

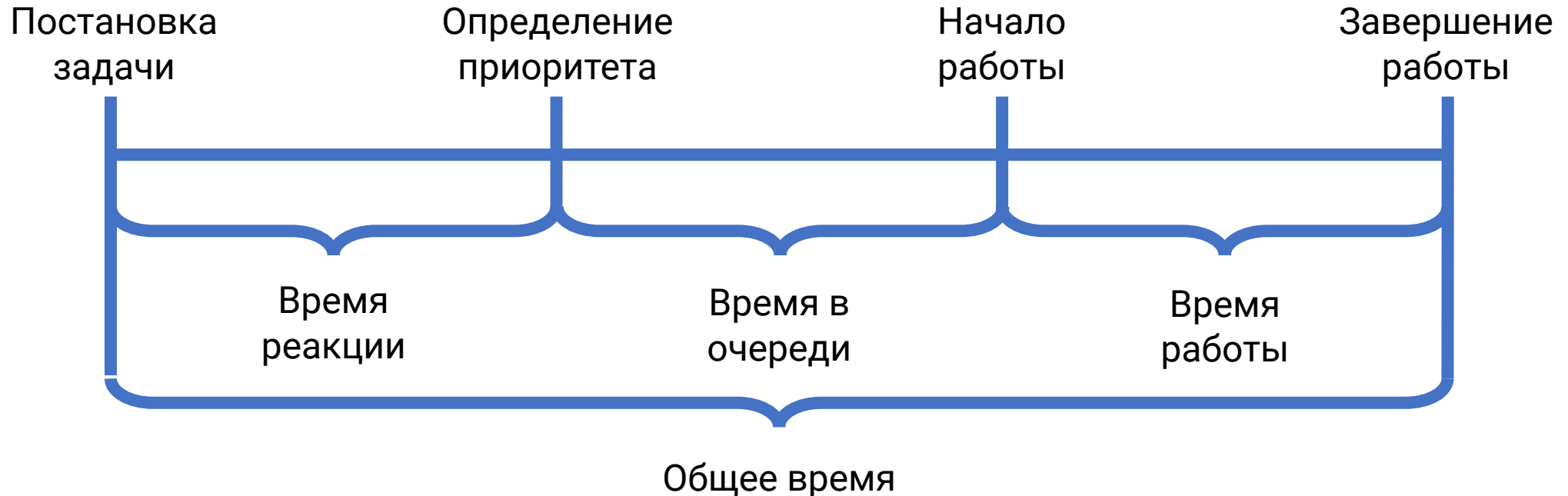


- Зачем нужны метрики?
- Быстро провалиться и не потерять деньги – это важно в инновационных проектах, где нужно проверить бизнес-гипотезу
- Идентифицировать риски на ранней стадии проекта
- Чтобы хорошо спланировать ваш проект и не оказаться на бирже труда:)

ПРИМЕР ИЗМЕРЕНИЙ



- Lead time – время между тем, когда задача была поставлена и тем, когда клиент увидел ее в конечном продукте
- Cycle time – время между тем, когда была начата работа над задачей и тем, когда клиент увидел ее в конечном продукте.



ПРИОРИТЕТЫ



МЕТОД MoSCoW



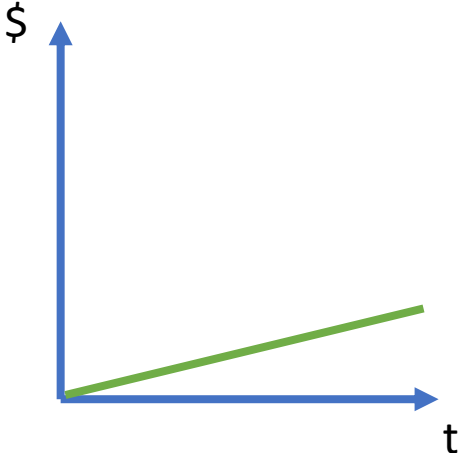
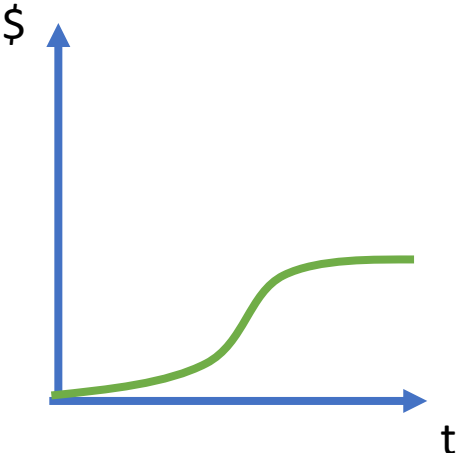
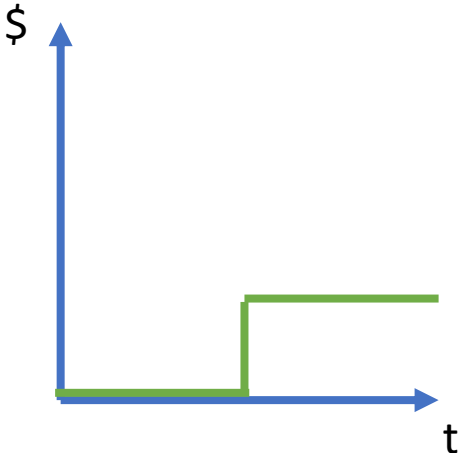
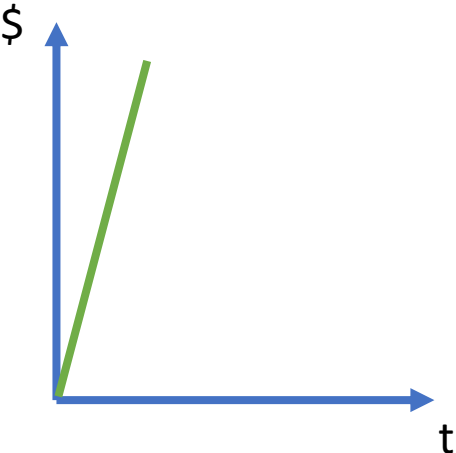
M – must – самый высокий приоритет, без которого продукт не будет сдан, проект не будет выполнен

S – should – важные требования, не имеющие решающего значения, но обязательные к исполнению

C – could – желательно сделать

W – would – наименее критичные требования

ПО СТОИМОСТИ И ВРЕМЕНИ



МЕТОД ФУТБОЛКИ

ПРИОРИТЕТ = ЦЕННОСТЬ - ЗАТРАТЫ



	Очень большие затраты (7)	Большие затраты (6)	Средние затраты (1)	Малые затраты (0)
Ценность очень большая (7)	0	1	6	7
Ценность большая (6)	-1	0	5	6
Ценность средняя (1)	-6	-5	0	1
Ценность малая (0)	-7	-6	-1	0

WEIGHTED SHORTEST JOB FIRST. СНАЧАЛА НАИВЕСОМЕЙШАЯ КРАТЧАЙШАЯ РАБОТА WSJF = ЦЕННОСТЬ/ЗАТРАТЫ



Название	Ценность, \$	Затраты, ч	WSJF	Номер в очереди
Задача 1	10000	160	63	10
Задача 2	20000	150	133	5
Задача 3	40000	300	133	4
Задача 4	50000	400	125	6
Задача 5	25000	70	357	1
Задача 6	1000	10	100	9
Задача 7	5000	20	250	2
Задача 8	100000	850	118	7
Задача 9	15000	150	100	8
Задача 10	35000	200	175	3

ЦЕЛЕПОЛАГАНИЕ



КАКИЕ ПРОБЛЕМЫ АКТУАЛЬНЫ?



Непонимание общих целей ограничивается команды в возможностях реализации собственного потенциала. Это делает их просто исполнителями

Разобщенность целей приводит к рассинхронизации работы подразделений или даже к внутрикорпоративной «войне». Левая рука не ведает, что творит правая.

Разные трактовки результата приводят к непониманию между отделами. Одни думают, что успех — это когда подписали договор с партнером. Другие, что это существенный рост заказов от нового партнерства.

Непрозрачность вызывает беспокойство у команд, стейкхолдеров, акционеров.

ИЗ ЧЕГО СОСТОИТ И КАК УСТРОЕНА ЦЕЛЬ В ФОРМАТЕ OKR?



OKR = Objectives & Key Results

O – objective, цель. Формулировка цели. Пример:
«Запустить новую версию продукта»

KR – key results, ключевые результаты, критерии достижения. Мы договариваемся, что считается достижением цели. Обычно ключевых результатов несколько штук.

СВОЙСТВА ЦЕЛИ И KR



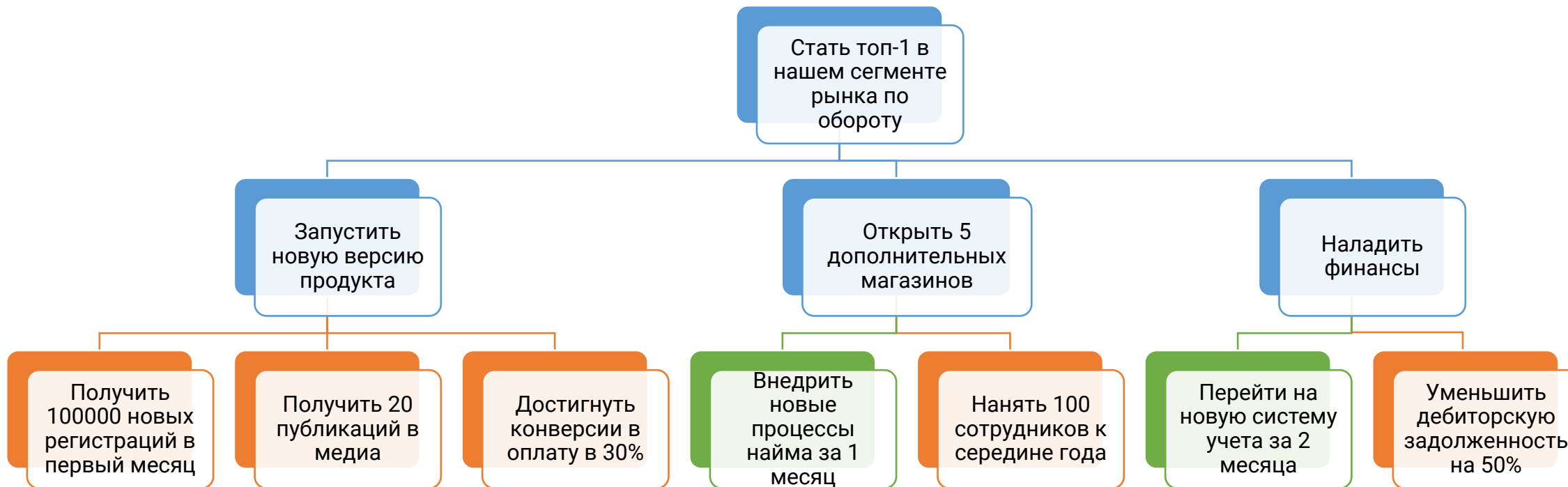
- Свойства цели
- Уточняющие вопросы
- Позитивная лексика
- Простота
- Начало с глагола
- Пример KR
- Ключевые, а не все подряд
- Результаты, а не задачи
- Позитивная лексика
- Просто и ясно
- Поиск наилучшего
- Назначение ответственных

СВОЙСТВА ОКР



1. Цели/Objective часто максимально амбициозные и труднодостижимые. Для таких выполнение KR на 80% - уже успех
2. Бывает цель «просто сделать». Ее KR надо выполнить на 100%. Ребрендинг или сделан в нужную дату или не сделан. Дом построен или нет.
3. 1-3 цели на команду, 2-5 ключевых результатов для каждой цели. Если больше, то это уже потеря фокуса.
4. Публичность – все сотрудники видят цели всех и каждого на каждом уровне. Это необходимо для поиска точек соприкосновения.
5. Вертикальная связность целей – фокусировка команды на действительно важных вещах. Формирование связанного дерева целей.
6. Измеримость результатов. В абсолютных (числа) или относительных (проценты) значениях.
7. Достижение ОКР и премирование – не связанные вещи.

ЦЕЛИ И КЛЮЧЕВЫЕ РЕЗУЛЬТАТЫ ОБРАЗУЮТ ИЕРАРХИЮ



OKR vs KPI vs MBO?



OKR

- Цели и ключевые результаты на квартал.
- Вначале отвечаем на вопрос, «зачем мы это делаем».
- Направление движения на коротком промежутке времени.
- Это про исследование, поиск новых путей и точек роста.
- Поэтому ключевые результаты по умолчанию очень высоки, иначе зачем все это затевать.
- Не используется для наказания и поощрения сотрудников
- Пример: О стать топ-1 в своем сегменте.

KPI

- Ключевые показатели эффективности, контролирующие ежедневную деятельность
- Явно достижимые показатели успешности деятельности, чаще всего понятной и налаженной.
- В них нет формулировки новой цели.
- Используется для наказания и поощрения сотрудников
- Примеры: EBITDA, Выручка, GMV YoY

MBO

- Цели на год
- Количество целей не ограничено
- Каскадирование сверху-вниз и снизу-вверх
- Формулировка по SMART
- Могут быть кросс-цели
- Пример: Привлечь 1,5 млн пользователей средней ценовой сегмент и обеспечить отношение расходов к доходам по сегменту не больше 24%.

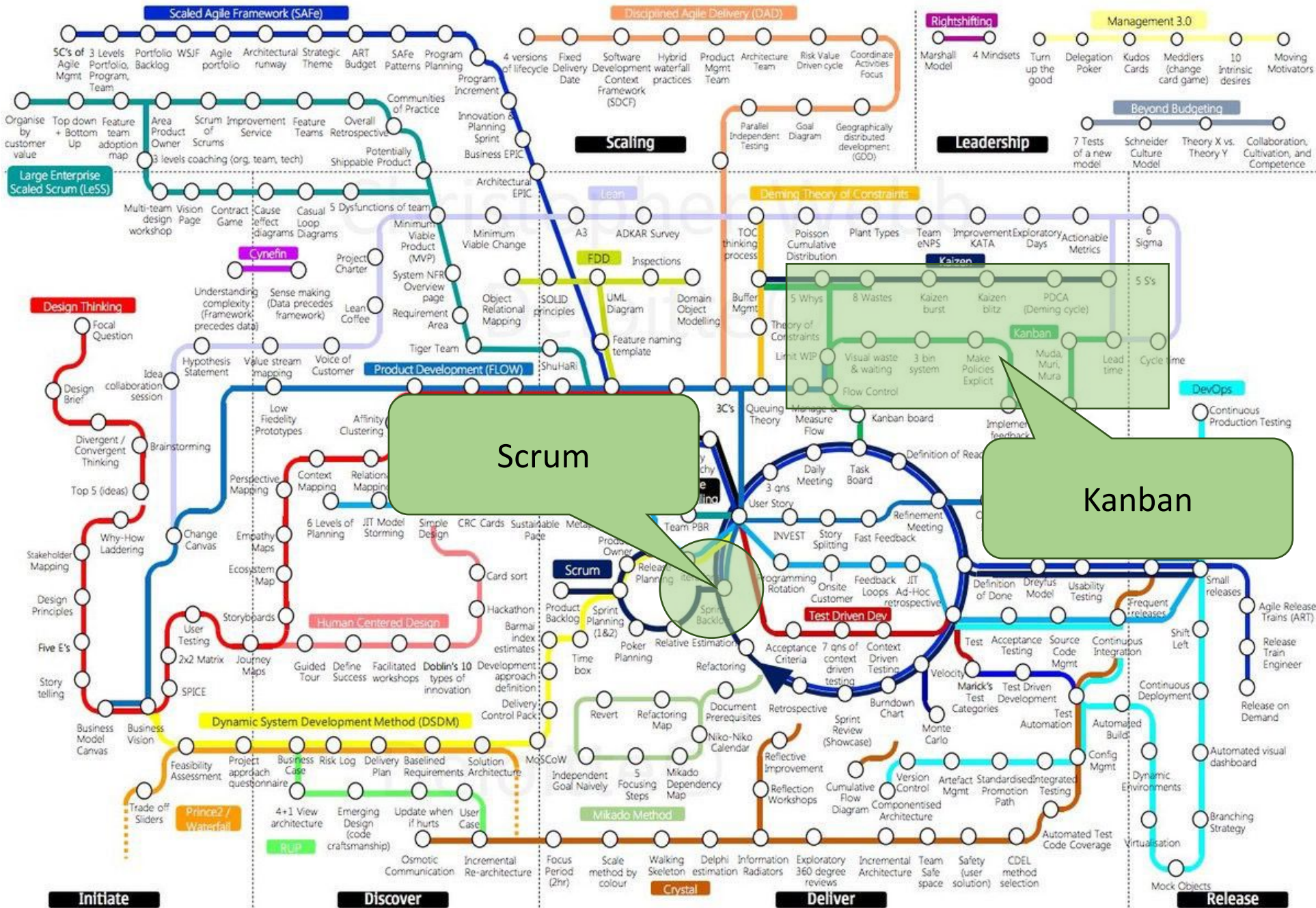
ЗАКЛЮЧЕНИЕ



ПОДВОДИМ ИТОГИ



1. Посмотрели на то, из каких предпосылок возникли ГМУП и определили их место в общем ряду подходов
2. Провели сравнительный анализ Agile и Lean, и наиболее популярных их представителей Scrum и Kanban
3. Поговорили немного про оценки, метрики, приоритеты и целеполагание в ГМУП



<http://blog.deloitte.com.au/navigating-the-agile-landscape/>

#ШКОЛАНОВАТОРОВ | Гибкие подходы к управлению проектами | Александр Колесников | 04.02 – 11.02



ВЫСШАЯ ШКОЛА БИЗНЕСА
ЦЕНТР КОРПОРАТИВНОГО ОБУЧЕНИЯ



TMX
ИНЖИНИРИНГ