

Прозрачность движения к результату в Agile для руководства



Максим Цепков

IT-архитектор и бизнес-аналитик,
навигатор и эксперт по миру Agile,
бирюзовых организаций и Спиральной динамике

<http://mtsepkov.org>

Конференция CodeFest
Новосибирск 31 марта – 01 апреля 2018

Мой опыт и знания

- Создание и внедрение больших корпоративных систем (более 20 лет)
 - Знание практик операционного управления и ведения проектов в крупных коммерческих и государственных организациях и банках
 - Опыт управления проектами в ИТ: от классического подхода (РМВОК) – к современным Agile-методам (с 2007)
 - Опыт перестройки процессов организаций при внедрении систем
- Сопоставление различных подходов управления
 - Agile и классический менеджмент (ряд [статьей и выступлений](#))
 - Модель развития [Сpirальной динамики](#) (с 2013)
 - Практики бирюзовых организаций от Фредерика Лалу и холакратия
 - СМД-методология, развитие СРТ в новой промышленной революции

Чтобы не было не оправданных ожиданий...

- В докладе не будет набора настроек и отчетов для Jira
- И даже не будет простого списка «что нам делать» и «какие метрики»
 - Потому что проекты – отличаются от других и команды – тоже
 - Потому что в каждом проекте – свои сложности, проблемы и вызовы
 - А если у Вас все одинаково – значит Вам не нужен Agile (может, его и нет?)
- В докладе будет
 - Сборка различных подходов, принципов и моделей
 - Применяя которые Вы построить систему, обеспечиваю прозрачность работы команд и движения проектов для руководителей компании и заказчика

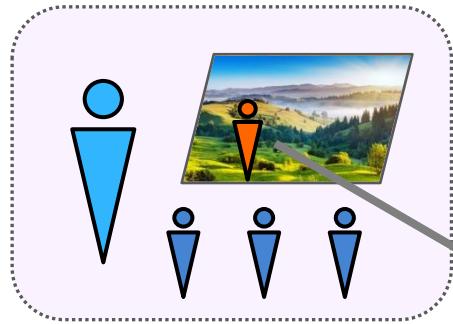


В готовых фреймворках Вы имеет собранную конструкцию «из коробки», а чтобы построить свою – надо понимать ее составляющие и способы компоновки

Доклад – карта местности, надо увидеть себя



Рассказ дает карту местности и возможные варианты движения



Необходимо увидеть на этой карте будущий путь своего проекта и себя на этом пути



Путешествие не является необходимым – это зависит от ситуации в проекте. Но пока ты не представишь себя идущим, материалы доклада будут мертвой теорией

По мотивам схемы из моего доклада «[Самоопределяйся технологично!](#)»

Прозрачность – проблема?

Почему возникает проблема прозрачности?

- Классический менеджмент основан на регламентах
 - Есть описанные процессы и правила, подходы – знакомы руководителям
 - Подразделения и проекты устроены одинаково
 - Поэтому у руководителей есть наработанные способы мониторинга
 - Даже если они действуют через коммуникации, а не KPI – они знают что спросить
- Agile – это новый, непривычный способ, а все команды – разные
 - Прозрачность работы одной команды есть, но требует прямой коммуникации
 - Наработанные способы наблюдения перестают действовать
 - Новые – не понятно, как вырабатывать, не ясны подходы и принципы
 - При этом каждый проект и команда движутся по своей траектории развития

А зачем вообще нужна прозрачность?



Разные цели –
разные решения

- Просто снять беспокойство руководителя
 - Беспокойство – обосновано предыдущим опытом менеджмента
 - И усиливается из-за высокой автономности команд в Agile
- Оказать экспертную помощь команде в решении проблем
 - Часто люди ищут решение, хотя существует готовое – они об этом не знают
- Вести мониторинг движения совокупности проектов и команд
- Оптимизировать организацию работы по компании в целом
 - Устранять проблемные места взаимодействия
 - Оптимизировать организацию структурно
 - Оценивать потенциал ресурсов для новых проектов

Снять беспокойство руководителя/стейкхолдера

- Идентифицируйте ситуацию: руководитель не хочет глубоко вникать, он просто хочет быть спокоен, что все идет «как надо»
- Поймите, каких фейлов и непредвиденных камней он опасается
 - В любом проекте могут случиться форс-мажоры, и не все они беспокоят
 - Что ему важно: сроки, качество, клиентские отношения, чтобы не было вины...
- Поймите язык, привычный руководителю и говорите на нем
 - [Презентация Dan Rawsthorne "Scrum: the Big Picture"](#) на AgileDays-2012 – как представить практики Agile на языке классического менеджмента
 - Разные руководители привыкли к разным моделям и языку организации работ



Отнеситесь к этому, как к части проектной работы: на некоторых наших проектах на это уходило 30-50% времени аналитика, это нельзя напрямую предъявить, но альтернатива – множество справок и совещаний по проекту

А можно ли взять готовое и не думать?

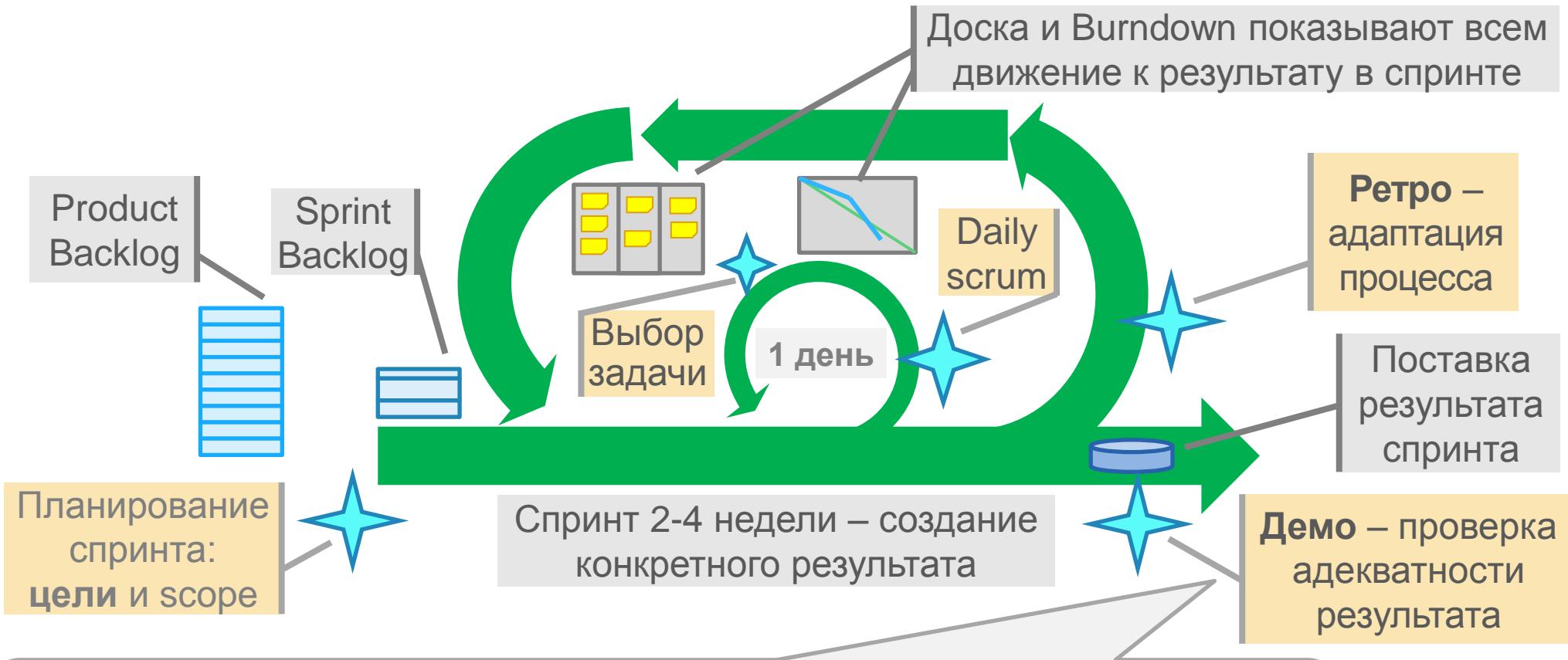
- При небольшом числе команд – можно освоить способы уровня команды, включенные в Agile-методы и регулярно их применять
- Можно использовать фреймворки уровня компании, в которые встроены способы мониторинга
 - [LeanKanban](#) – Kanban в масштабе компании с петлями обратной связи для взаимной настройки сервисов
 - [Scaled Agile Framework](#) (SAFe) – сложная конструкция уровня компании с управлением потоками создания ценности и архитектурой
 - [Enterprise Scrum](#) (автор Mike Beedle) – переход от создания IT-продукта к поставке ценности, управляемой набором связанных метрик
- Но это требует перестройки всей компании или всех подрядчиков, и будет ограничивать гибкость и возможности развития



Отдельная компетенция –
оценивать проект по KPI

Прозрачность одной команды

На уровне Scrum команды – все OK ?

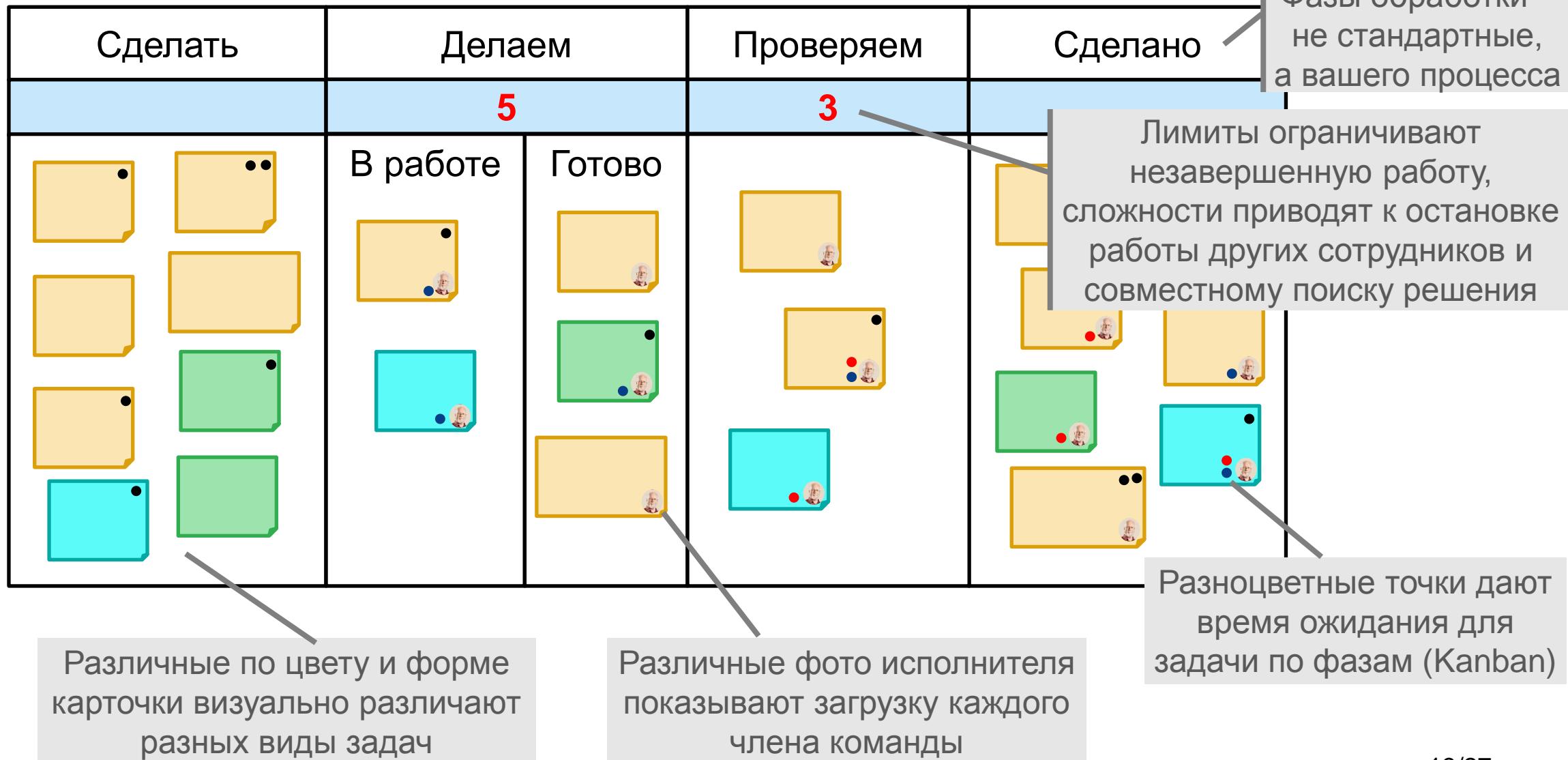


Проблем не возникает, только если в спринте действительно создается законченный ценный для потребителя продукт, а не просто выполняется определенное количество задач. И Вы можете получить на демо адекватную обратную связь

Ретро и Демо – обратная связь по спринту

- Простой вариант: продукт есть кому оценить, а команда – зрелая
 - Демо – обратная связь по продукту от стейкхолдеров и пользователей
 - Ретро – оценка командой спринта и выработка улучшений
- Не хватает обратной связи стейкхолдеров по работе команды
 - Расширяем Демо до Sprint Review, Ретро – внутреннее мероприятие
- Если пользователей и стейкхолдеров нельзя позвать на Демо
 - Неверный вариант: провести Демо для Product Owner – его оценки недостаточно
 - Надо: конструировать сложную коммуникацию, чтобы получить обратную связь
 - Если мы делаем продукт для массового потребителя, то там надо работать с гипотезами о ценности, и проверять – получилось или нет
- Ретро: что нам мешает быть лучше и как это изменить?

Доска – визуализация потока задач



Физическая доска и таск-трекер

- Доска и burndown chat показывают продвижение команды в спринте
- Таск-трекер обеспечивает
 - Мониторинг и ретро-анализ на длинных горизонтах – если задачи вести аккуратно
 - Мониторинг и сопоставление разных проектов и команд – при однородном ведении
- Сохранять ли физическую доску, если есть таск-трекер – решайте сами
 - Для коммуникации команды и самоуправления физическая доска – полезнее
 - Можно использовать электронную доску или дублировать на обоих
- Можно ли без таск-трекера на большом масштабе? Можно, но не нужно
 - Представляем движение в рамках компании на Kanban-доске проектов
 - Показатели каждого спринта заносим в Excel и строим графики



Некоторые люди по-прежнему читают только бумажные книги, а общаются только по почте и лично, и говорят, что так им эффективнее. Это их выбор: не учиться новому, а работать привычными способами

Сделать задачу – создать ценность?

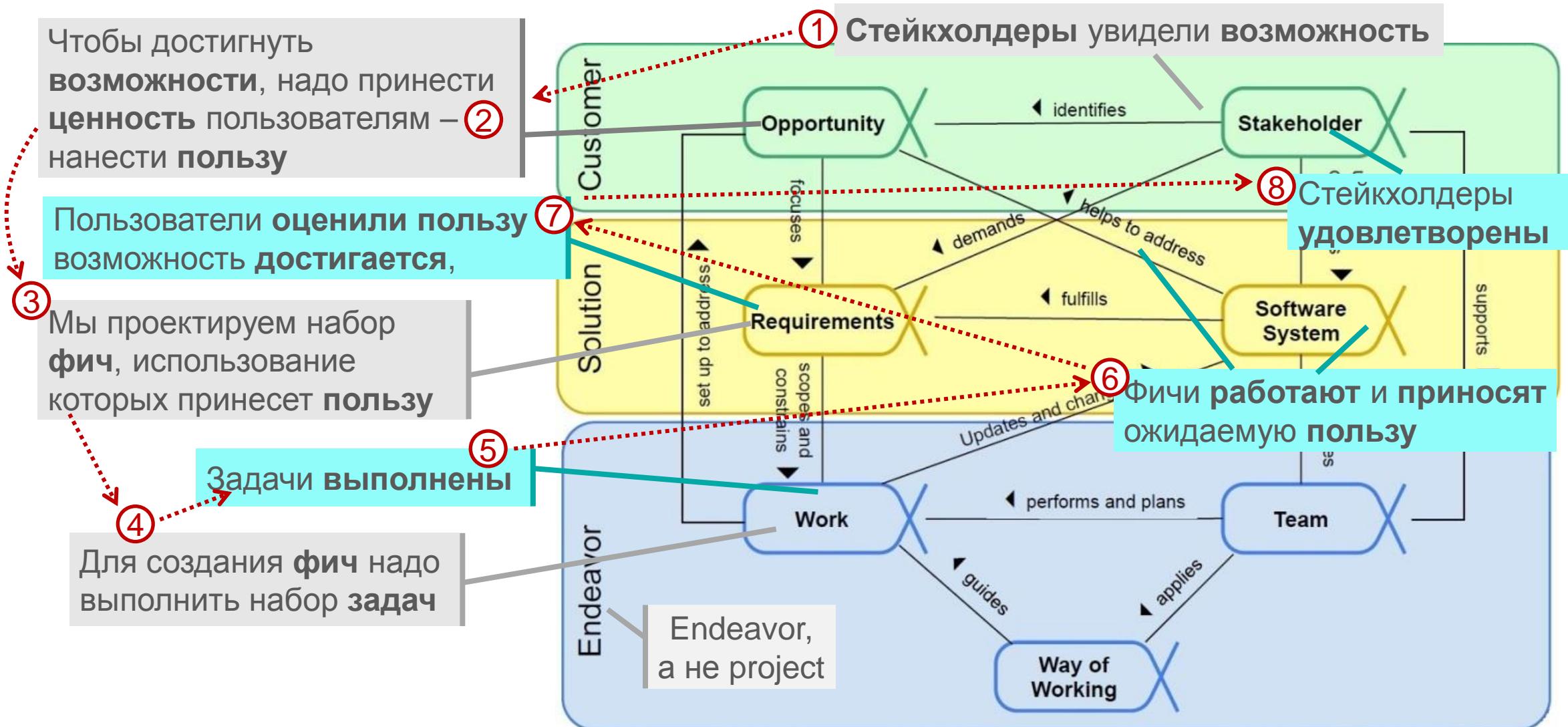
Task stream и Value stream

- Product Backlog Item – фича и ее ожидаемая польза – создание value
- Создание фичи может быть декомпозировано на задачи, именно их мы делаем в операционной работе
- В демо – проверка, получилось ли сделать ожидаемое, и могут появиться новые задачи, достраивающие целостность value
- Для понимания продвижения проекта надо отслеживать не только выполненные задачи, но и созданную ценность, необходимо поддерживать трассировку задач до создаваемой ценности

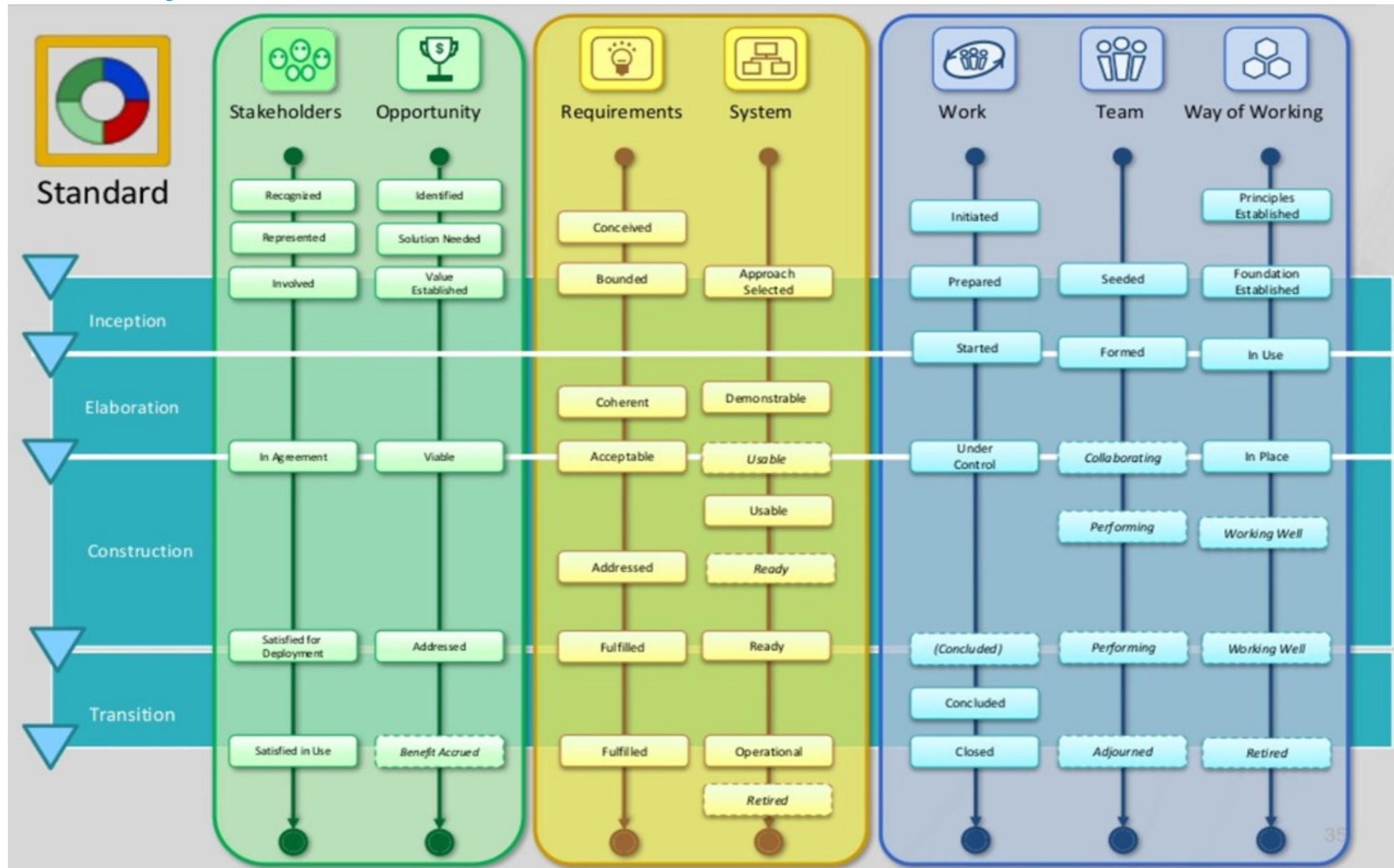


В проектном менеджменте между освоением бюджета и получением ценности – разрыв, и он связан с мировоззрением: мы все хорошо спланировали – как же результат может не принести ценность?

Ценность и работы на карте OMG Essence



Lifecycle OMG Essence – прогресс по всем альфам



Показатели value stream и мониторинг по ним

- Поток задач (затраты) в единицах оценки: скорость прохождения по фазам, точки задержки потока, объем сделанного, оценка не сделанного, прогноз исполнения, расходование буфера проекта
- Ценность для пользователя: использование новых фич, инциденты и обратная связь по ним от пользователей
- Достижение возможностей: новые пользователи, денежный поток, расширение по сегментам рынка, факт против ожиданий
- Удовлетворение стейкхолдеров: новые задачи и проекты от них, отсутствие претензий и инцидентов, хорошее взаимодействия
 - Оценивать можно по-разному, экспертно и статистически
 - У разных стейкхолдеров – разные интересы, их надо удовлетворять по разному: одни продолжают двигать проект, а другие – просто не мешают движению

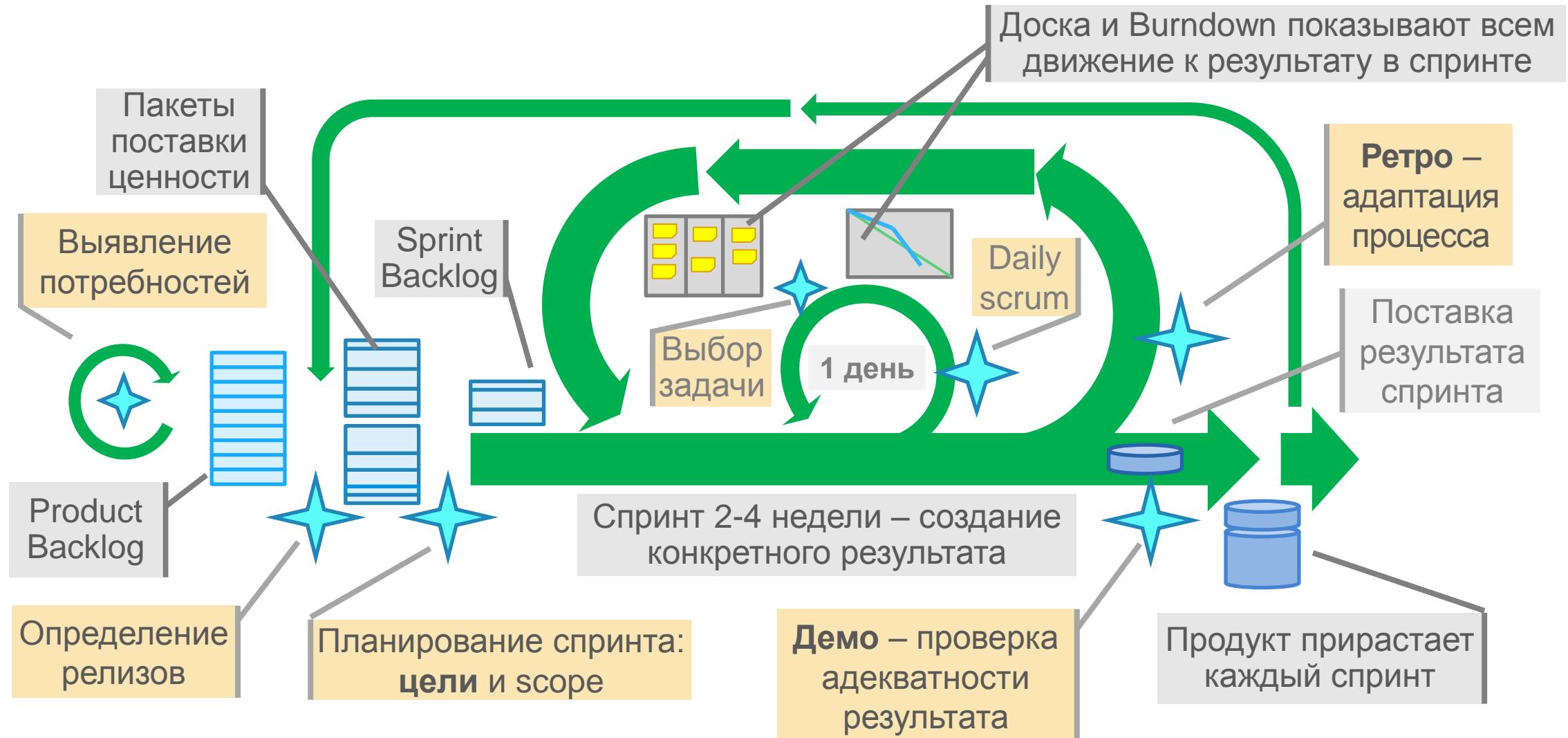
Мониторинг по показателям – компетенция

- Показатели не дают решения, они указывают точки внимания
 - Этому хорошо учат показатели рынка и использования фич: нельзя дать команду
- Показатели надо превратить в **индикаторы**, найдя границы, выход за которые требует внимания – светофорная модель
- Для мониторинга нужно визуализировать на графиках
 - Но, как и с таск-трекером, средства для графиков – не главное
 - А вот удачный набор показателей, постоянно видимых на мониторе – важен



Компетенция мониторинга по показателям – редкая компетенция. Характерна ситуация с мотивацией по KPI, многие видят ложную альтернативу: или автомат зарплаты и премий по KPI, или неформально обсуждаем без них

Квантование потока – релизы в Scrum



Механизмы управления проектом

- Формулируем целостные релизы с точки зрения потребителя
 - Даже для внутренних проектов – они доставляют ценность
- Используем выделение Minimum Viable Product (MVP)
 - Не только для первого релиза, но и для последующих
 - Ориентация релизов на группы пользователей и группы функций
- Управляем проектом через важность и приоритеты



Поток создания ценности не всегда можно поделить на содержательные релизы, он может состоять из слабо зависимых задач. Тогда можно использовать Kanban. Или все равно использовать Scrum – он лучше передает mindset Agile, а спринты можно использовать для планирования изменений и совершенствования процесса

Мониторинг многих проектов и продуктов



Поток создания ценности не зависит от процессов, он описывает работу и agile-команд и подразделений под классическим управлением

Методологии работы с потоком ценности

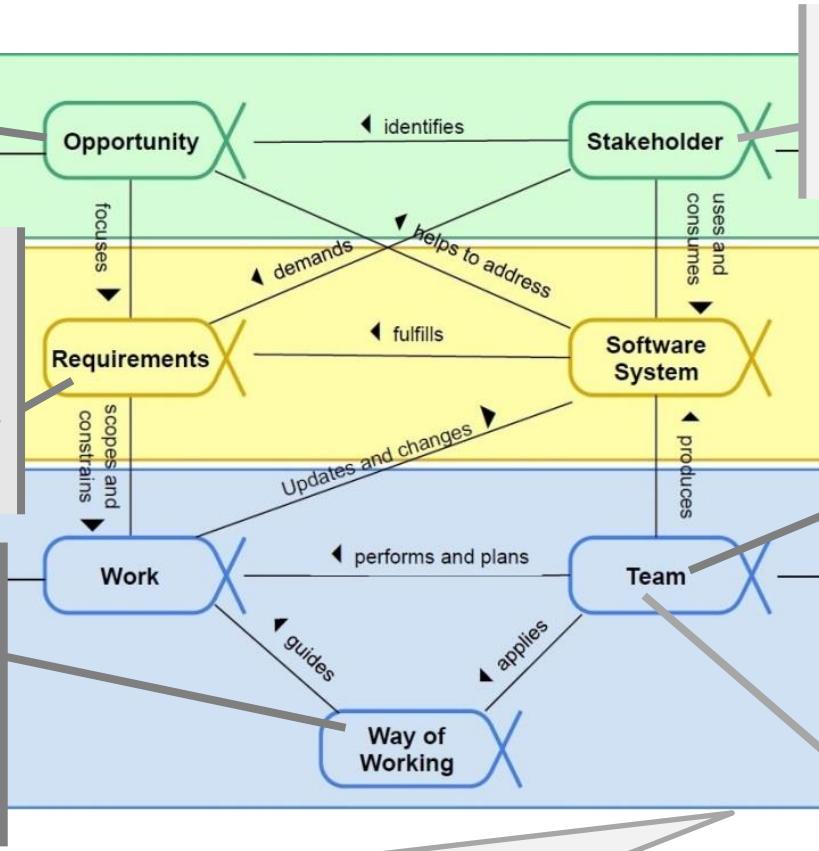
- Концепция есть в Lean, Value Chain и ТОС Голдратта, они были адаптированы к ИТ (Мэри Поппенчик, Дэвид Андерсон)
- Enterprise Scrum, LeanKanban, SAFe предлагают варианты KPI
- В готовых наборах показателей нет конкретной специфики компании – их необходимо адаптировать
- Концепт гибкой работы с бюджетированием – Beyond Budgeting bbrt.org
 - [Видео доклада Bjarte Bogsnes](#) (Vice President Statoil, Норвегия) на [IT-Spring-2017](#)

Цепочки создания ценности в компании

- Простая компания: каждый проект/команда создает конечную ценность
 - Можно суммировать и сравнивать между собой
 - Учитываем фазы жизненного цикла продукта – на них разная динамика
- Модель Адизеса зрелой корпорации: дойные коровы, зрелые бизнесы, молодые развивающиеся проекты
- Длинные цепочки создания ценности
 - Ядро продукта, разработка на нем под разных заказчиков
 - Общая серверная часть и много клиентских решения (web, мобилки, десктоп) – движение по лидеру, по основному сегменту, по отстающему
- Сервисная модель компании: различаем структуру и инфраструктуру
 - Инфраструктура – сервисы, которые не ограничивают движение проектов за счет собственной избыточной мощности, показатели для нее – другие

Индикаторы проблем – для конкретных ситуаций

- Новые фичи не используются
- Пользователи недовольны
- Пользователи пассивны
- Демо и/или внедрение показывает не адекватность
- Неожиданно много workaround
- Много возвратов на разработку
- Слишком дорогая разработка
- Проблемы в типовых задачах
- Нестандартное – супердорого
- Разная скорость команд на одинаковых задачах
- Долгое освоение новичками



Стейкхолдер

- постоянно требует объяснений
- блокирует новые инициативы

Разработчик

- мало делает
- плохо продумывает задачи
- не держит фокус выпуска
- перегружен

Команда

- накапливает тех.долг
- много незаменимых
- не видит ценность работы
- работа не интересна



Был доклад на TeamLead Conf от разработчиков GitLean
[Оцениваем разработчика на основе объективных данных](#) – там раскрыта механика, как сделать индикаторы проблем.
А ТОС Голдратта выявляет проблемы для потоков

Подводя итоги

- Надо учиться и учить работать с метриками: готовые средства есть, но надо уметь их использовать
- Подбирайте и экспериментируйте с метриками проекта, есть много докладов, где рассказывают о своих системах метрик
- Визуализируйте необходимые показатели и индикаторы
- Метрики показывают проблемы, но не дают способ работы – это повод для разговора, а не готовое решение
- Поймите причины беспокойства руководителей и отвечайте на них



Максим Цепков
<http://mtsepkov.org>
maks.tsepkov@ya.ru

На сайте много материалов по [анализу и архитектуре](#), [Agile](#), и [ведению проектов](#), [управлению знаниями](#), мои [доклады](#), [статьи](#) и [конспекты книг](#).