

Развитие управления проектами и критериев качества в ИТ

Максим Цепков

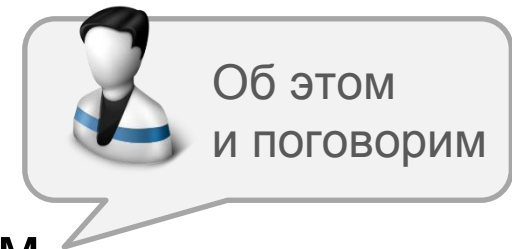
Главный архитектор дирекции развития решений

Москва, 19 марта 2015 года



О чем этот доклад

- ▶ Способы ведения проектов и представления о качественном результате регулярно меняются
- ▶ Это популярная тема холиваров
- ▶ У каждого свои представления:
 - Одни используют то, чему научили когда-то
 - Другие кропотливо накапливают личный арсенал
 - Третьи следуют модным трендам
- ▶ Все методики и практики формировались в своем контексте и уместны для конкретных видов проектов



План рассказа

- ▶ Исторический обзор
- ▶ Современные тренды
- ▶ Big Picture ведения проектов
- ▶ Применение на практике

История моды ведения проектов



Эпоха НИОКР: когда компьютеры были большими

- ▶ Квалифицированный персонал
- ▶ Большие и сложные проекты
- ▶ В которых редко менялись требования
- ▶ А упор был на качество решения

Ф. Брукс
«Мифический
человеко-месяц»



Были успехи и поражения – как в любом НИОКР

Появились персоналки

- 😊 Вау, можно автоматизировать каждую компанию!
- 😞 Но где взять столько квалифицированных разработчиков?
- 😊 А вроде и средненькие справляются...
- 😞 Только надо поставить процессы и регламенты

Эпоха RUP

- ▶ **ИТ-разработка как проект создания системы:** спроектировать, разработать и внедрить
- ▶ **Решение:** разделим задачу на этапы, создадим процесс их прохождения
- ▶ **Оценка качества:** по тому, удалось ли выполнить проект в срок, бюджет и с ожидаемым результатом

Да, много накладных расходов, зато результат гарантирован

PMBOK-3 (2004)
RUP (2003)

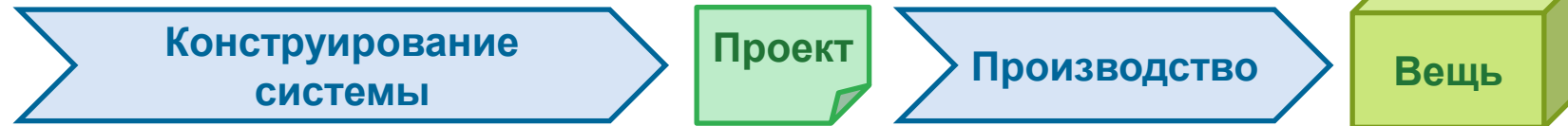


Фиг он
гарантирован...

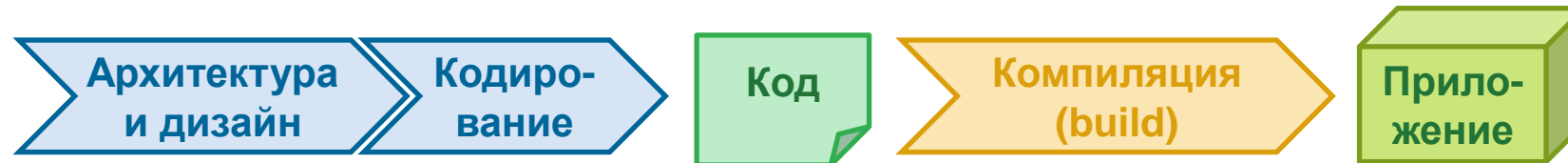
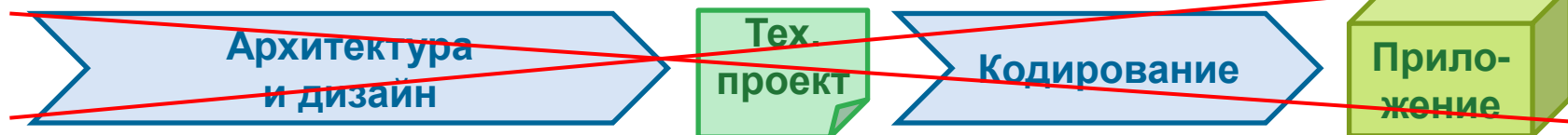
Природа ИТ мешает процедурам

[Jack W. Reeves «What is software design»](#) (1992; [перевод](#))

Обычный НИОКР



ИТ-разработка



Цена ошибки невелика, поэтому пробуем, отлаживаем, доводим – так дешевле.
Пока проект не становится слишком сложным

Вызовы, на которые не ответили

- ▶ **Стоимость:** процедура увеличивает ее кратно, не сильно повышая вероятность успеха
- ▶ **Изменчивость:** потребности меняются быстрее, чем проходит цикл разработки, и нужно учесть эти изменения
- ▶ **Управленческие кадры:** где брать, особенно руководителей групп?
- ▶ **Нормирование аналитической работы:** в РМВОК-4 попробовали – не получилось

В стандарте признано

Итерации в RUP – тяжелые

Agile и SCRUM: ответ на вызовы

- ▶ Вместо тщательного планирования – наблюдение за траекторией движения проекта и приближением к цели
- ▶ Концепция SMART-целей, измеримость достижения
- ▶ Итеративное движение с корректировкой положения цели (требования к системе)
- ▶ Оценка качества ведения проекта по адекватности оценки расстояния до цели и движения в итерацию

Гибкость
и наблюдаемость



Факторы успеха SCRUM

- Сохранилась доля успешных проектов
- Стало намного дешевле, чем «по RUP»
- Появилась возможность вносить изменения в ходе проекта
- Снизились требования к руководителям групп и команд
- Постепенно произошло масштабирование на большие проекты



В стандарте игнорировать не могли, включить – не получилось. В PMBOK-4 (2008) добавили итерации – вышла эклектика

Это – в мире. А в России?

- ▶ В 90-х ученые массово пошли зарабатывать деньги
- ▶ Это позволило долго держать НИОКР-способ разработки в ИТ
 - Нормирование процессов использовали слабее
 - SCRUM был не столь востребован, шел почти 7–8 лет: появился в начале 2000-х – пришел в конце 2000-х
- ▶ Сейчас различие нивелируется
 - Agile приходит в in-house

Что меняется сейчас?


Вектора развития

- От проектной деятельности – к непрерывному развитию продукта
- От качества ИТ-системы – к удовлетворенности стейкхолдеров
- От создания системы – к достижению возможностей для бизнеса и пользователя
 - Особенно в новых направлениях – стартапы, мобильная и массовая продуктовая разработка, игры
- Каждому проекту – свой метод работы

Канбан в ИТ (2010)
DevOps (2012)

PMBOK 5 (2013)
частично



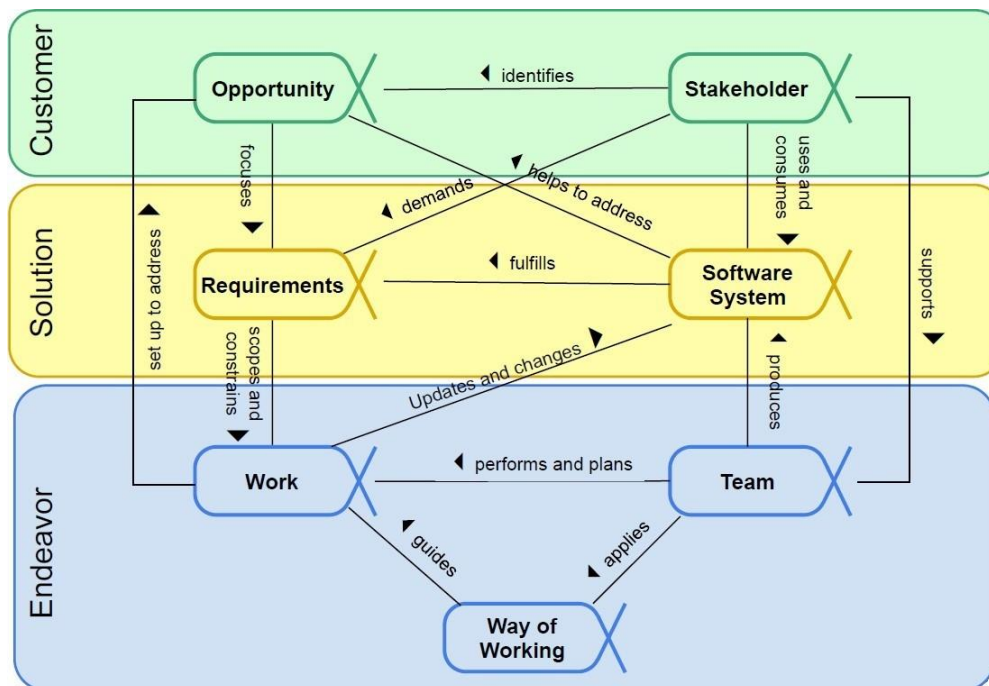
 Все это требует новых подходов...

Разработан SEMAT
(Ивар Якобсон)

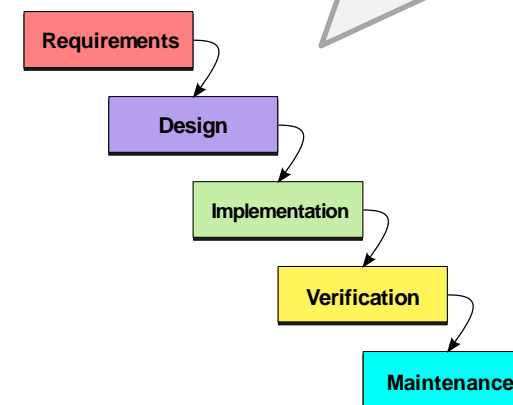
CUSTIS®

OMG Essence

- ▶ Простота – must!
- ▶ Из стандарта вычищено слово «проект»
- ▶ В альфах – стейкхолдеры и возможности



Предыдущая такая схема – водопад Ройса



А что с проектированием?

- 😊 В несложных или небольших системах – успешно, есть много практик
- 😞 А в сложных люди – ключевой фактор
- 😐 Лучшие решения **для сложных систем**
 - Процесс – **FDD** (Джефф де Люка)
 - Способ – **DDD** (Эрик Эванс)
- 😞 Оба – тяжелые

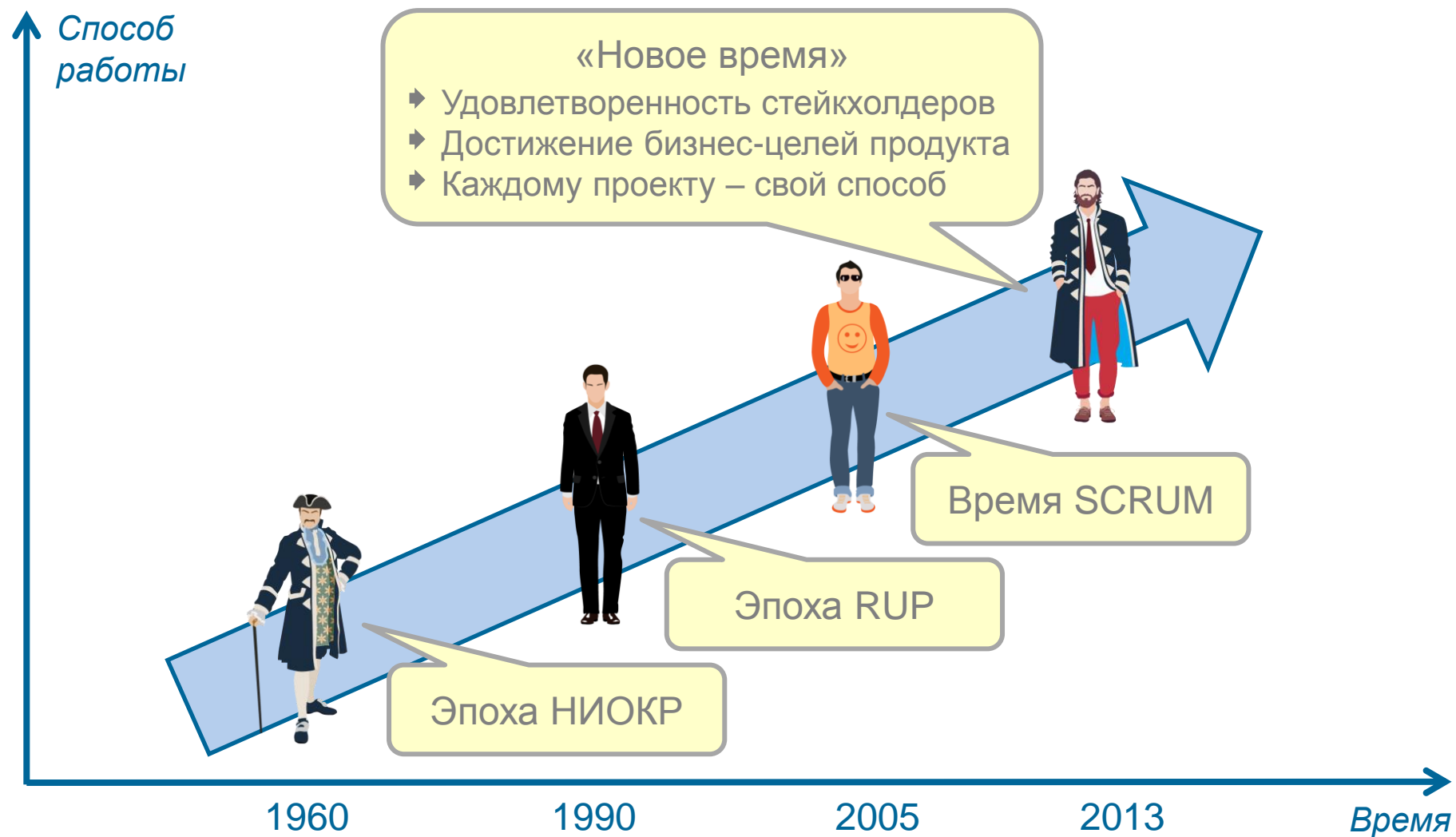


~~От истории – к действию~~

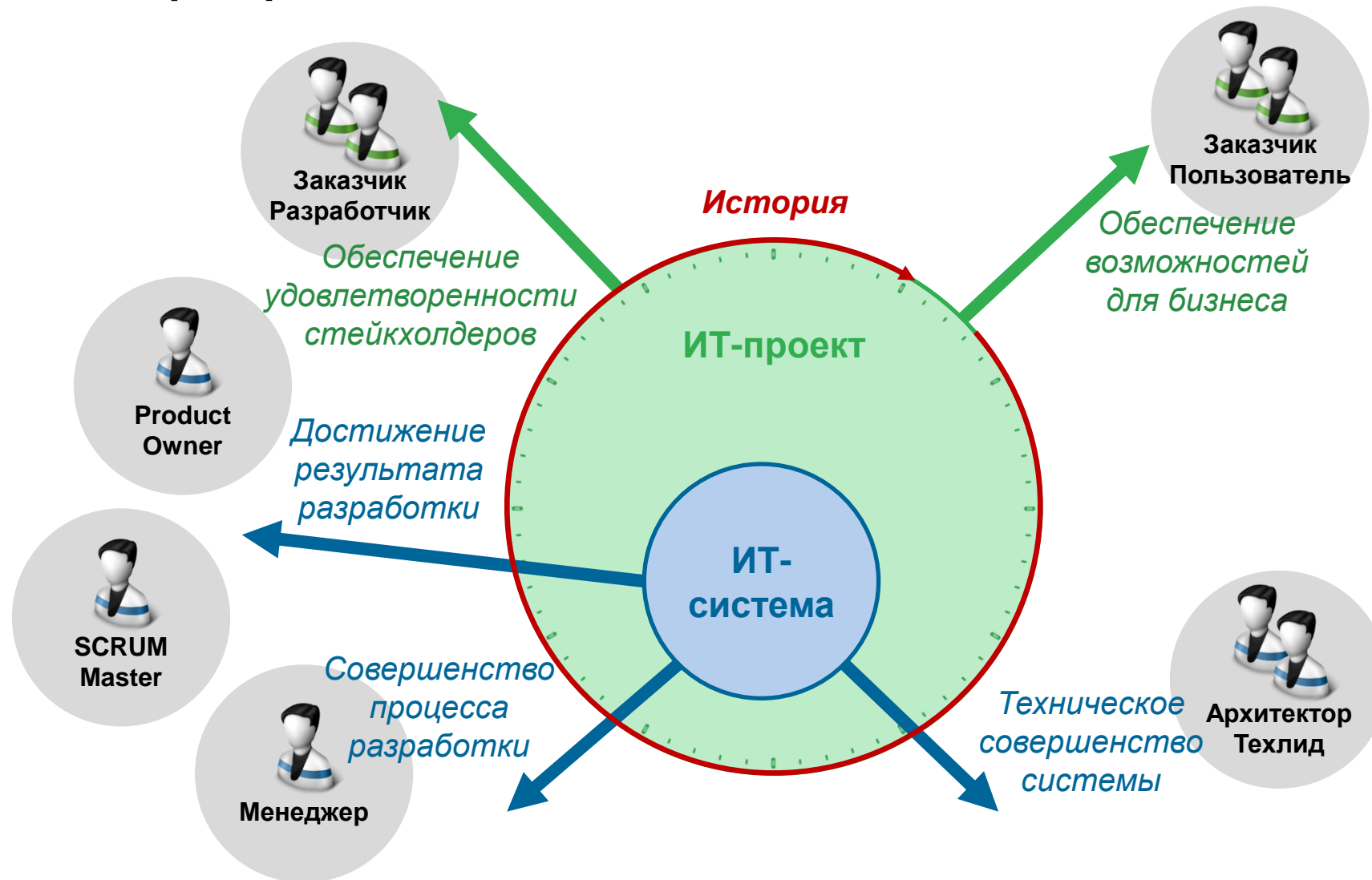
История ⇒ Модель ⇒ Действия



Big Picture истории развития



Вектора развития



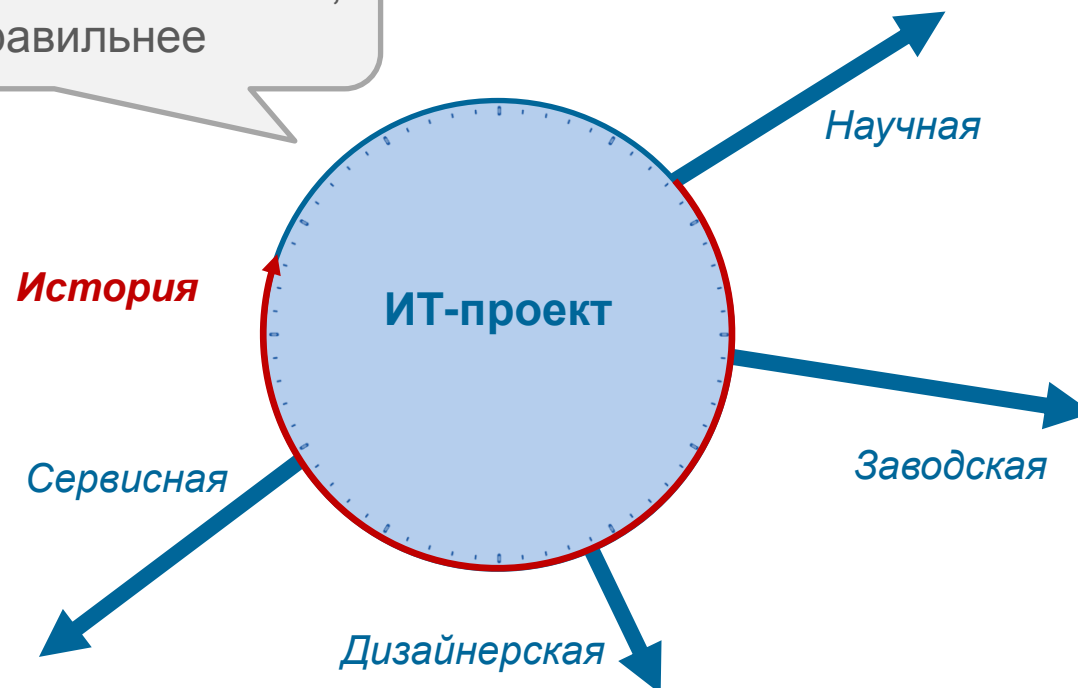
Энтони Лаудер

Культуры программных проектов

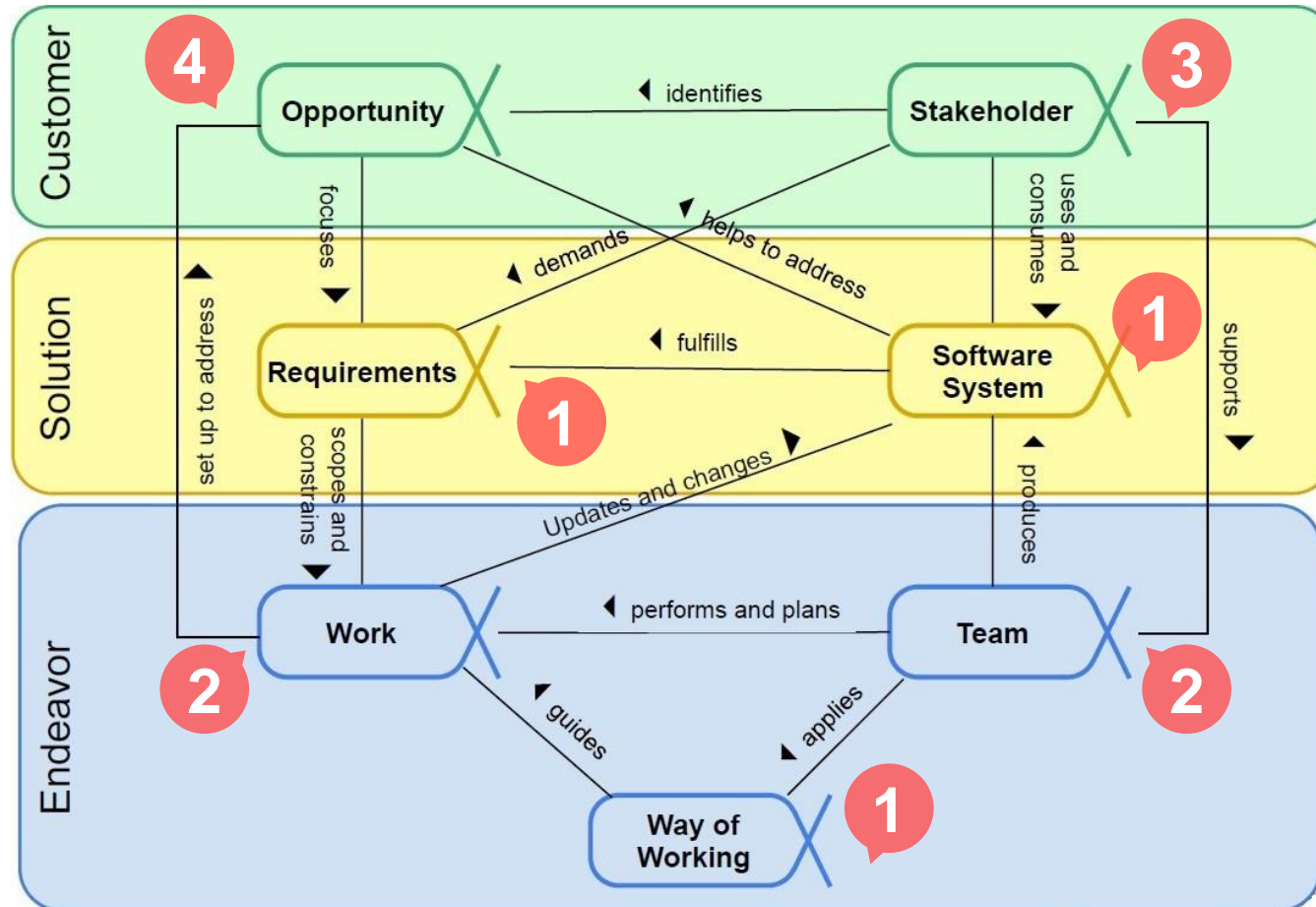


Моя схема мне подходит больше. Но это не значит, что она правильнее

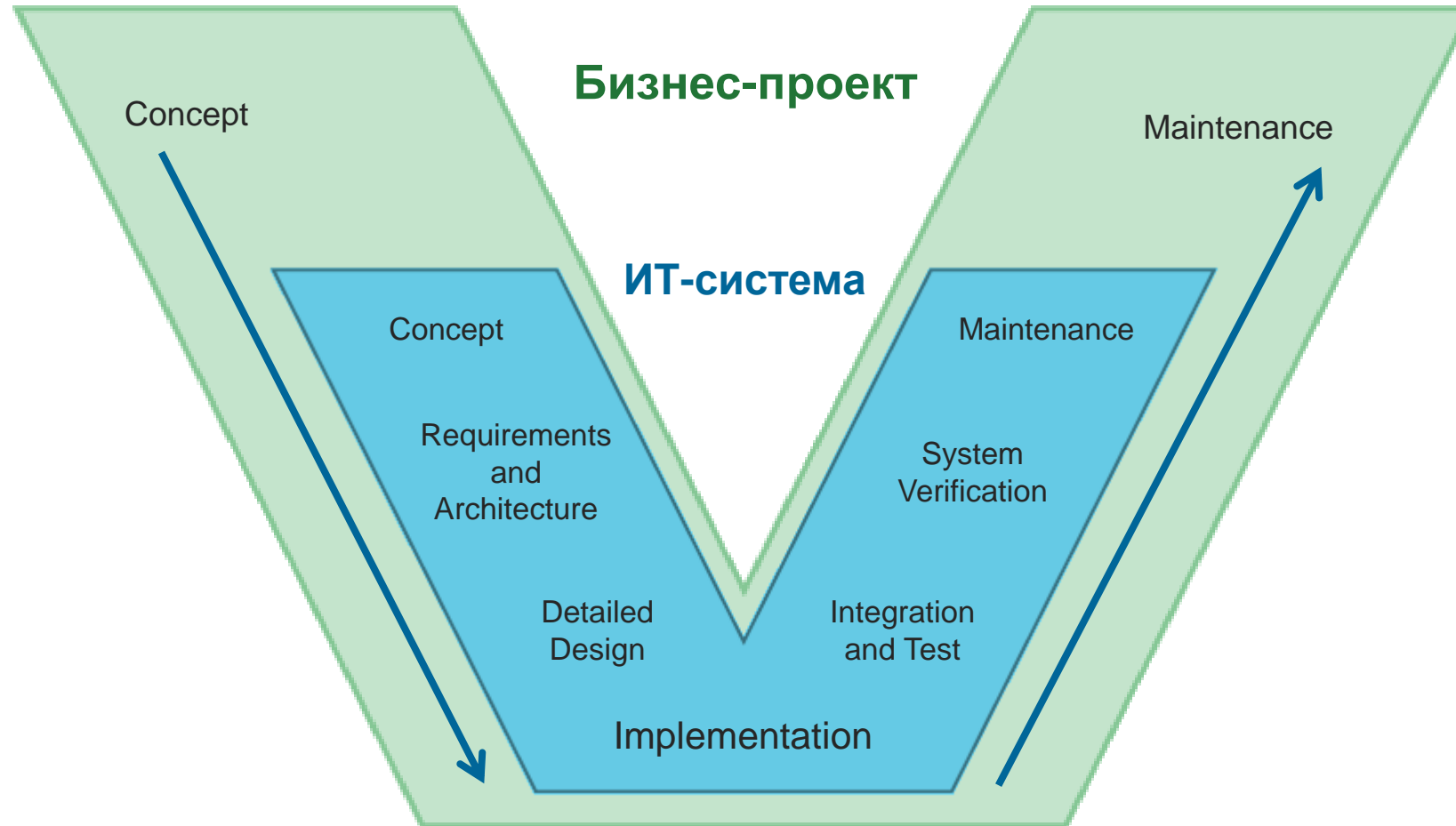
[Оригинал](#), [перевод \(pdf\)](#), [рецензия Стаса Фомина](#)



Развитие глазами OMG Essence



Изменения на V-диаграмме



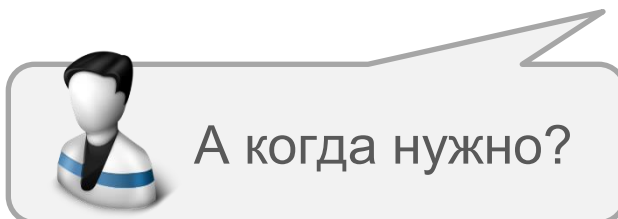
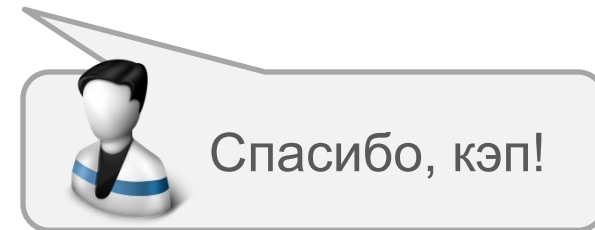
Расширение, а не отрицание

- ▶ Каждый следующий этап развития включает предыдущие, а не заменяет
- ▶ Но значимость предыдущих ценностей уменьшается: они перестают иметь исключительную важность

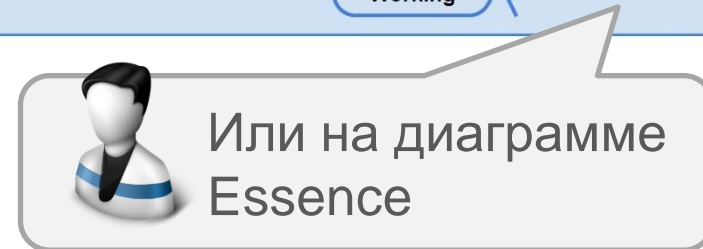
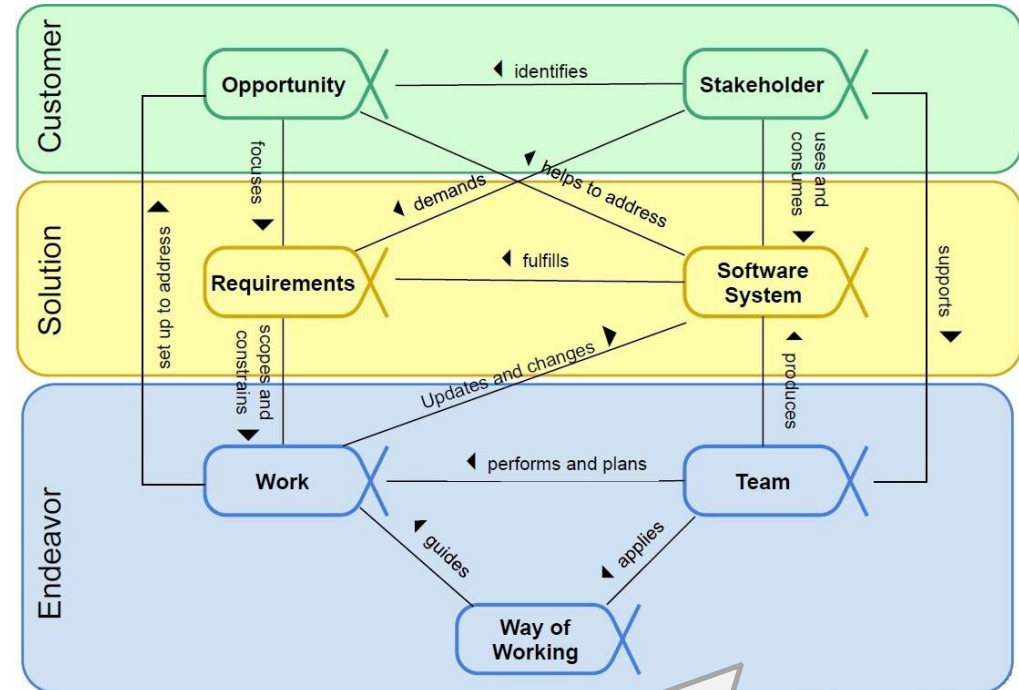
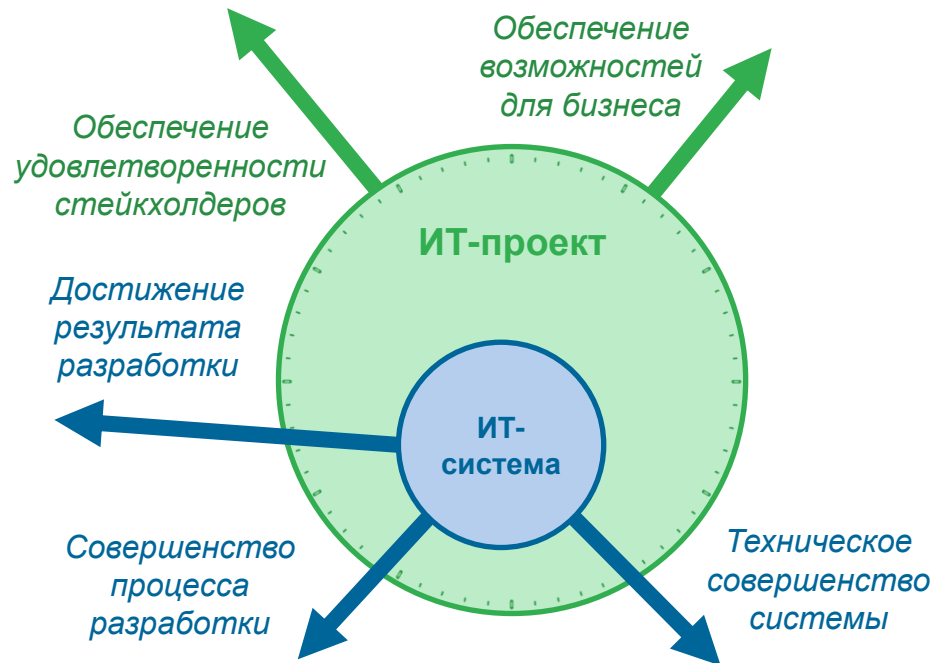
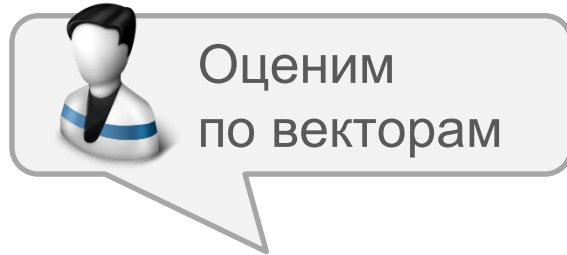
Модели есть –
можно применять на практике

Ценности для людей различны

- ▶ Представления о «правильном» способе ведения проекта и «правильном» результате у разных стейкхолдеров разные
 - У представителей заказчика
 - Менеджеров
 - Разработчиков...
- ▶ Нет задачи привести всех к одному мнению
- ▶ Но надо знать представления, а когда нужно – объяснять, работать с ними



В чем фишка проекта?



А как работает компания?

- Бизнес-модель
- Подходы к ведению проектов
- Найм персонала и работа с ним
- Манипуляции или сотрудничество?



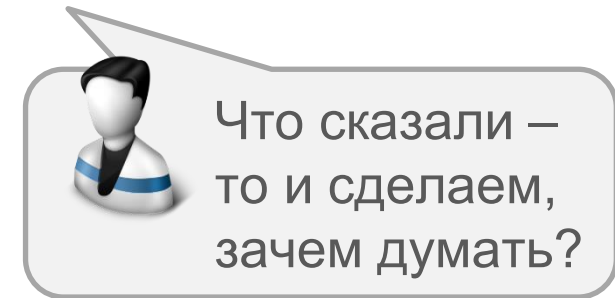
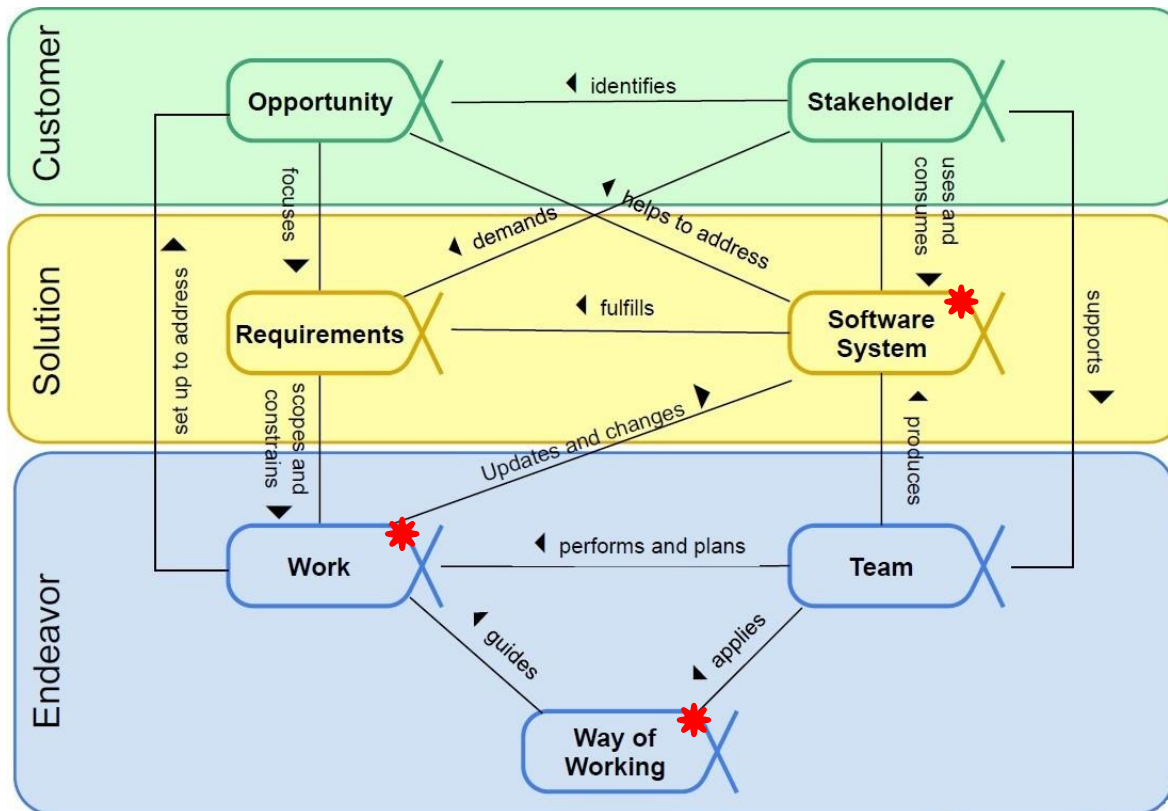
Что и как компания делает для мира?



Можно использовать **те же** модели – векторную, Essence и другие

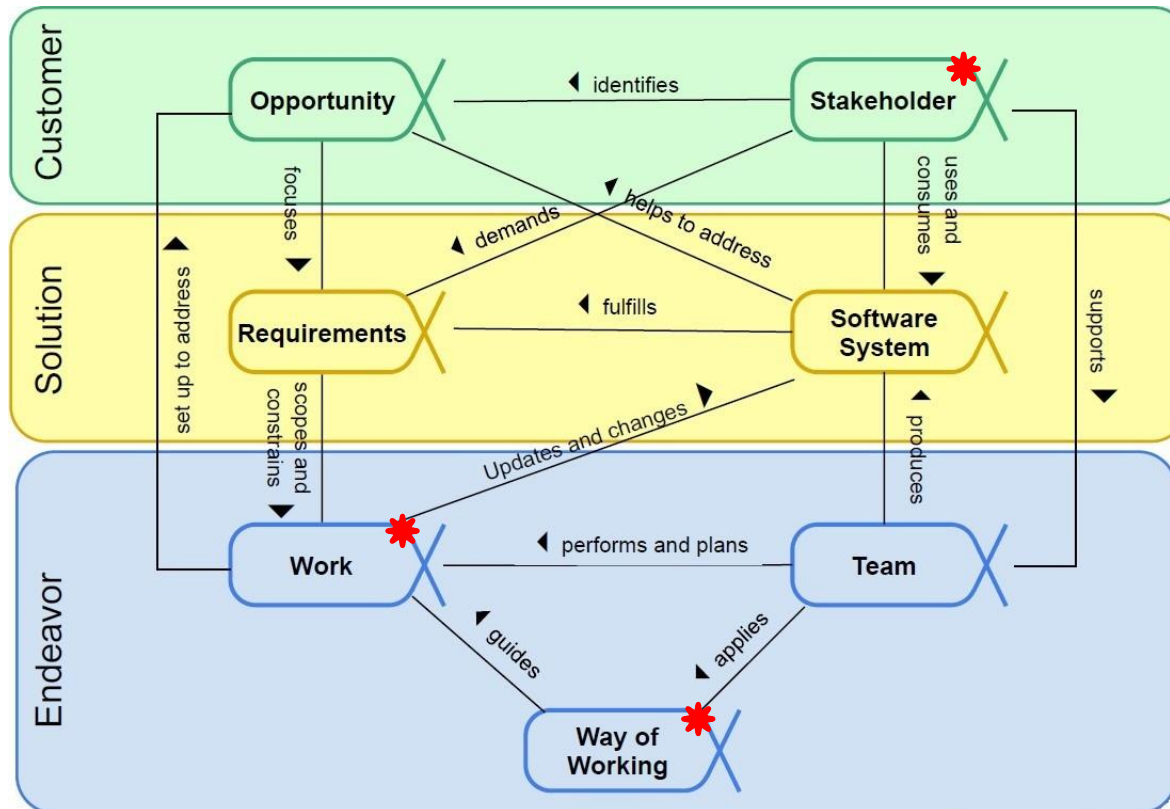
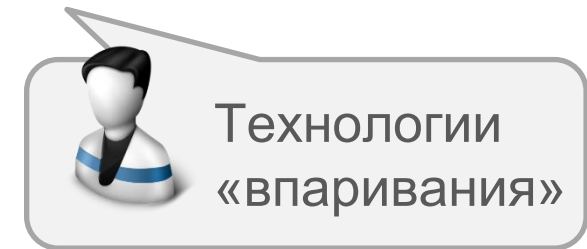
Разработка по спецификациям

Разрабатываем по строгим спецификациям доступным персоналом.
 Технологии обеспечивают приемку спецификации, декомпозицию работ на типовые с выполнением и сборку со сдачей заказчику по процедуре



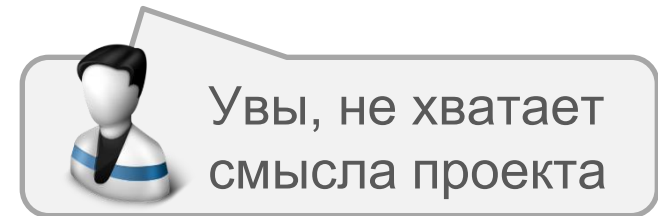
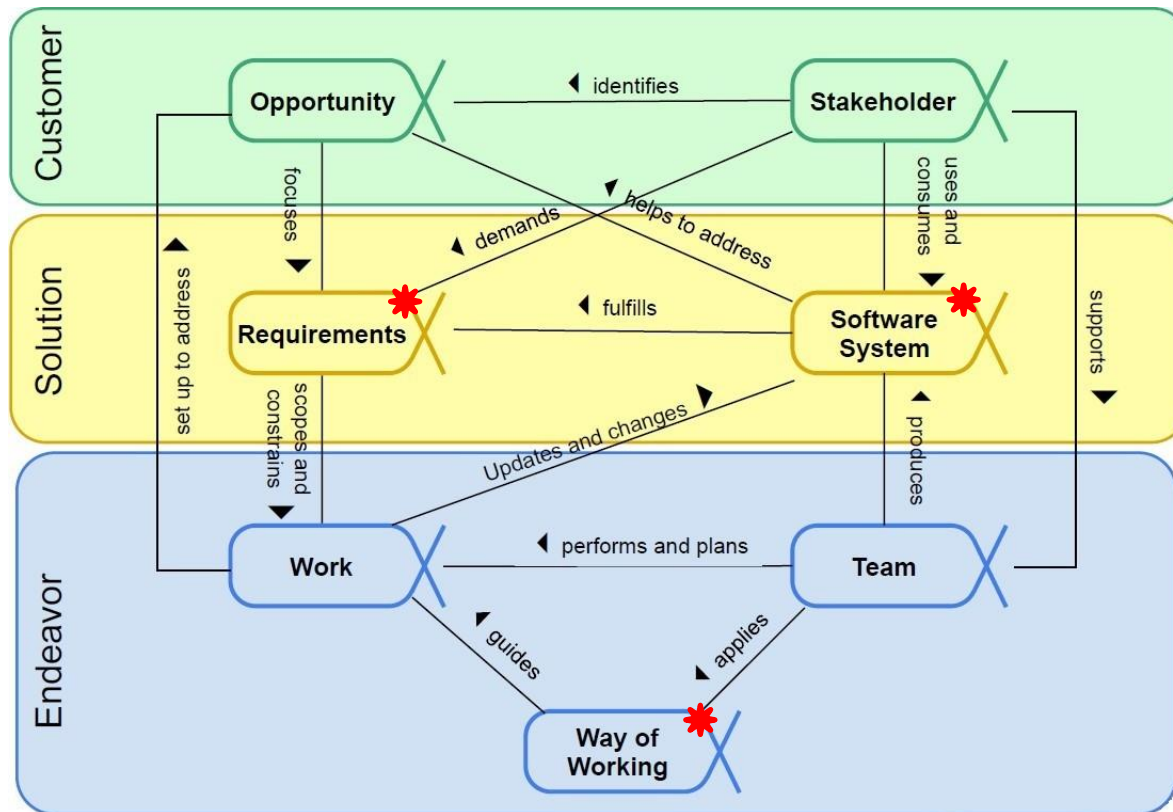
Продажа аутсорсинга разработки

При продаже обещаем качественный продукт, а потом обеспечиваем приемку того, что получилось сделать доступным персоналом, с дальнейшими доработками за отдельные деньги



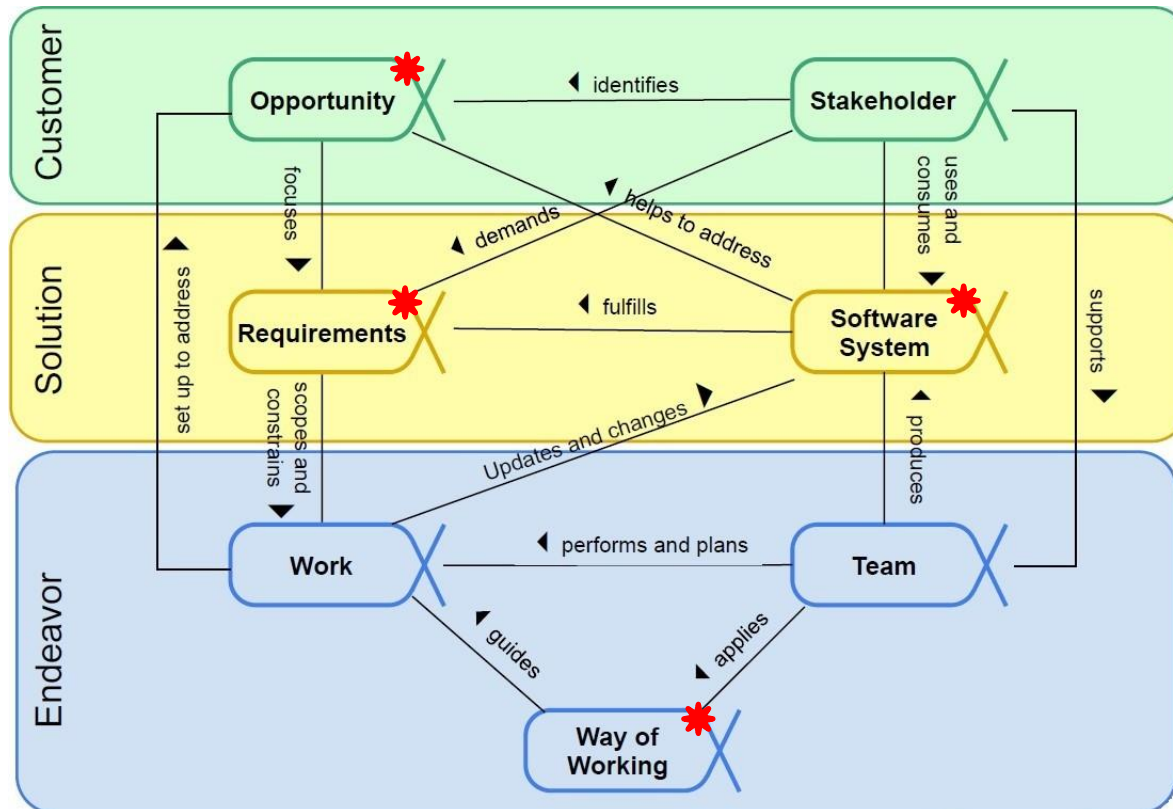
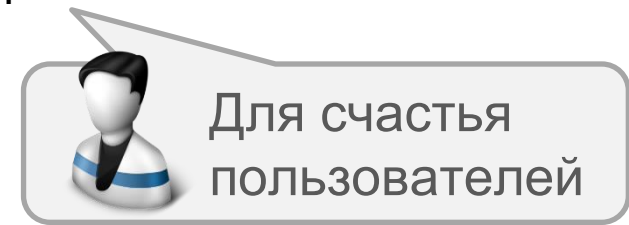
Создание качественных решений

Создаем совершенные высокотехнологичные системы посредством тщательного проектирования и воплощения



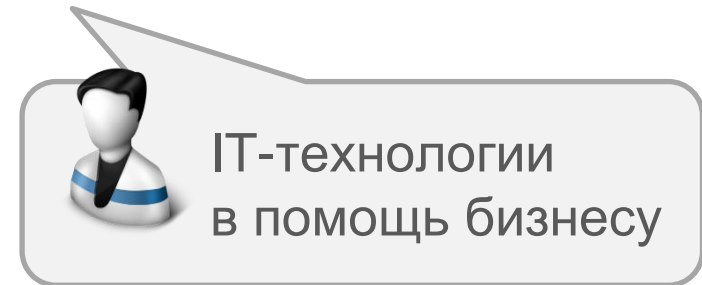
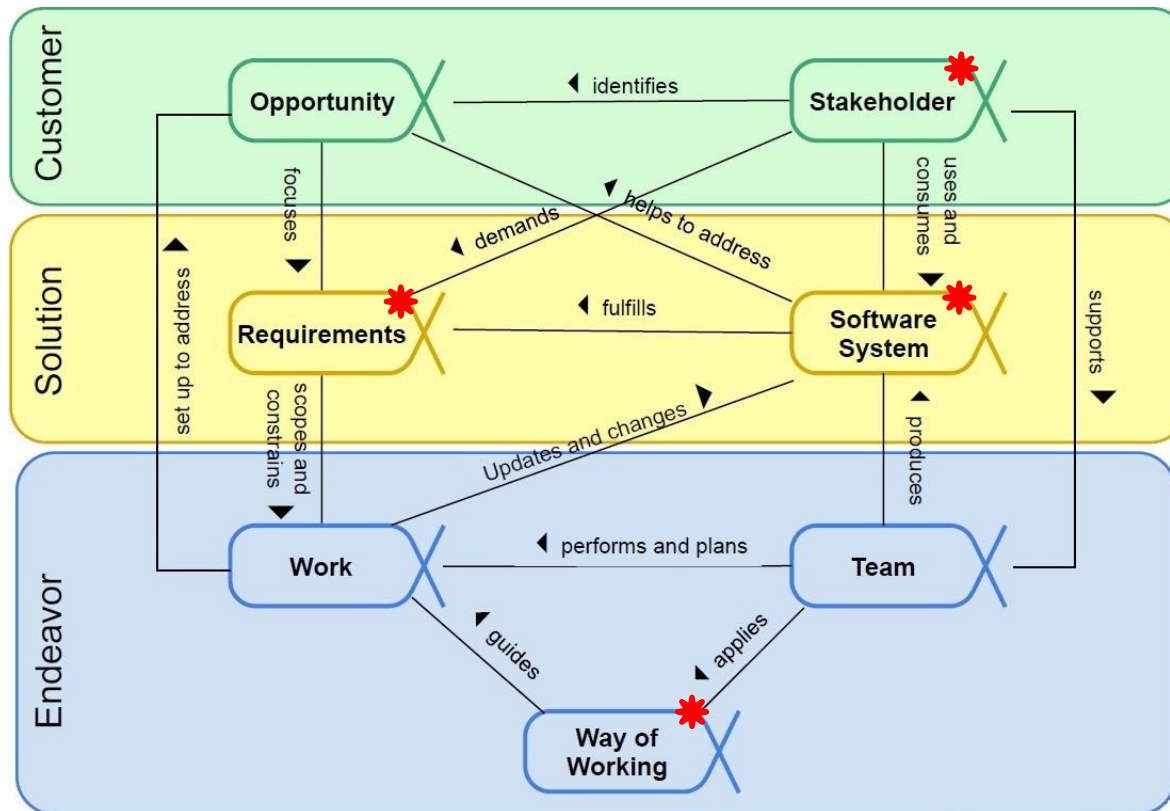
Высокотехнологичный стартап

Создаем высокотехнологичную систему, дающую пользователям принципиально новые возможности. Технологии обеспечивают не только разработку, но и работу с возможностями



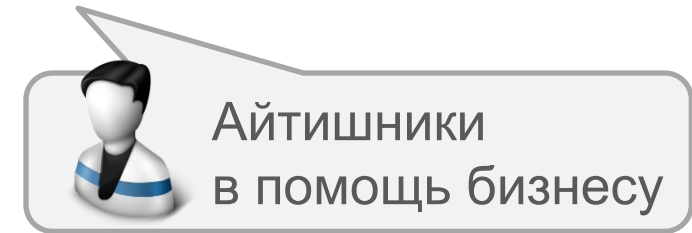
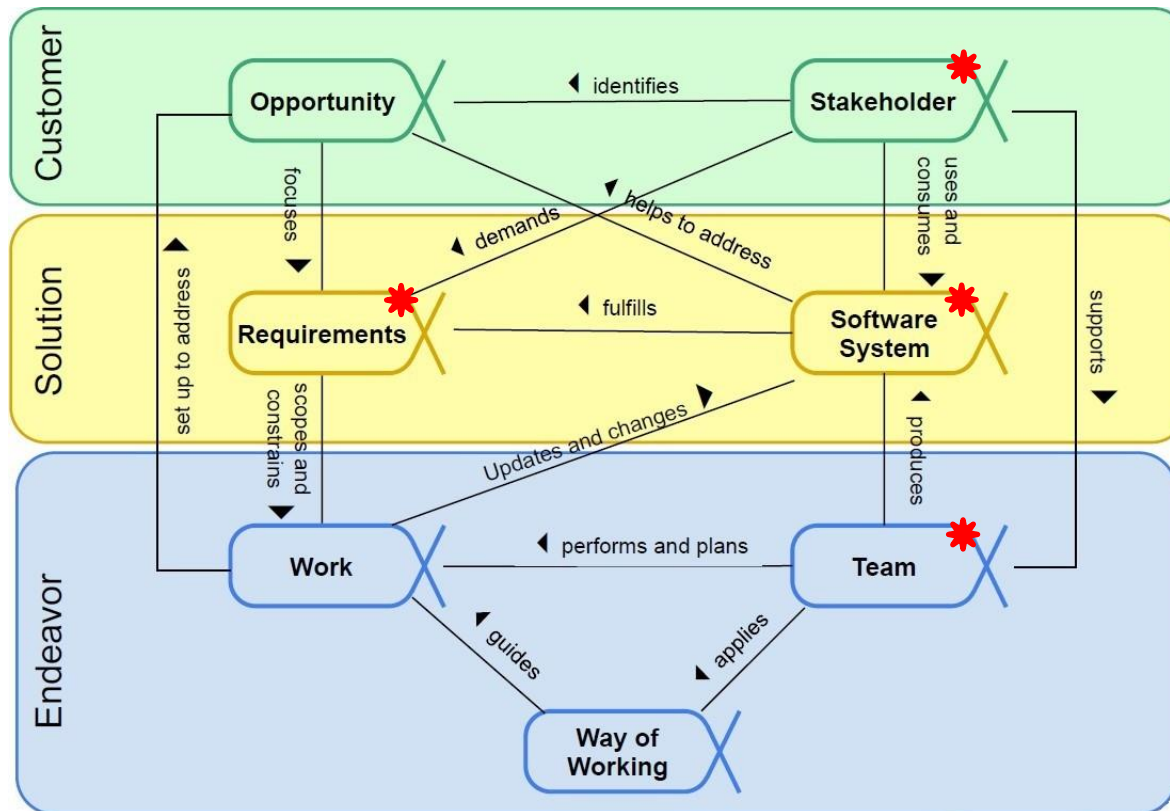
Технологичные системы для бизнеса

Создаем сложные системы, обеспечивающие решение проблем бизнеса. Технологии обеспечивают проектирование системы, отвечающей потребностям стейкхолдеров во взаимодействии с ними



Решение проблем бизнеса

Квалифицированная команда обеспечит разработку ИТ-систем, поддерживающих и обеспечивающих решение текущих задач бизнеса



Нужно ли понимать это каждому?

- ▶ Холиварные темы
 - «Глупые пользователи недовольны мелкими багами»
 - «Разработчики всегда делают что-то суперсложное»
 - «Аналитики идут на поводу у безумных пользователей»
- ▶ Надо понимать, в чем «фишка» проекта и фирмы, за что платят деньги
- ▶ И как твоя работа дает вклад в общее дело
- ▶ Хотя проектированием всегда занимается ограниченное число людей

Разобраться не сложно

- Есть формальные модели
- Есть шаблоны и методики
- Используем готовое и комбинируем



Модели для «неформальных» областей

- Стейкхолдеры и их цели
 - [ArchiMate Motivation Model](#)
 - [Модель описания целей i*](#) (i-star)
- Возможности – язык бизнес-стартапов и Minimum Viable Product (MVP)

Подводя итоги

Важно понимать



- ▶ Культуры ведения проектов и исторический контекст их возникновения
- ▶ Критерии успеха вашего проекта
- ▶ **Способ** работы вашей компании
- ▶ Имея для всего этого **модели**

Это дает бОльшую **осознанность деятельности**



Спасибо! Вопросы?

Максим Цепков mtsepkov.org