

Роли в проекте: как поделить поляну ответственности?



Максим Цепков

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

<http://mtsepkov.org>

Lead/Manage IT

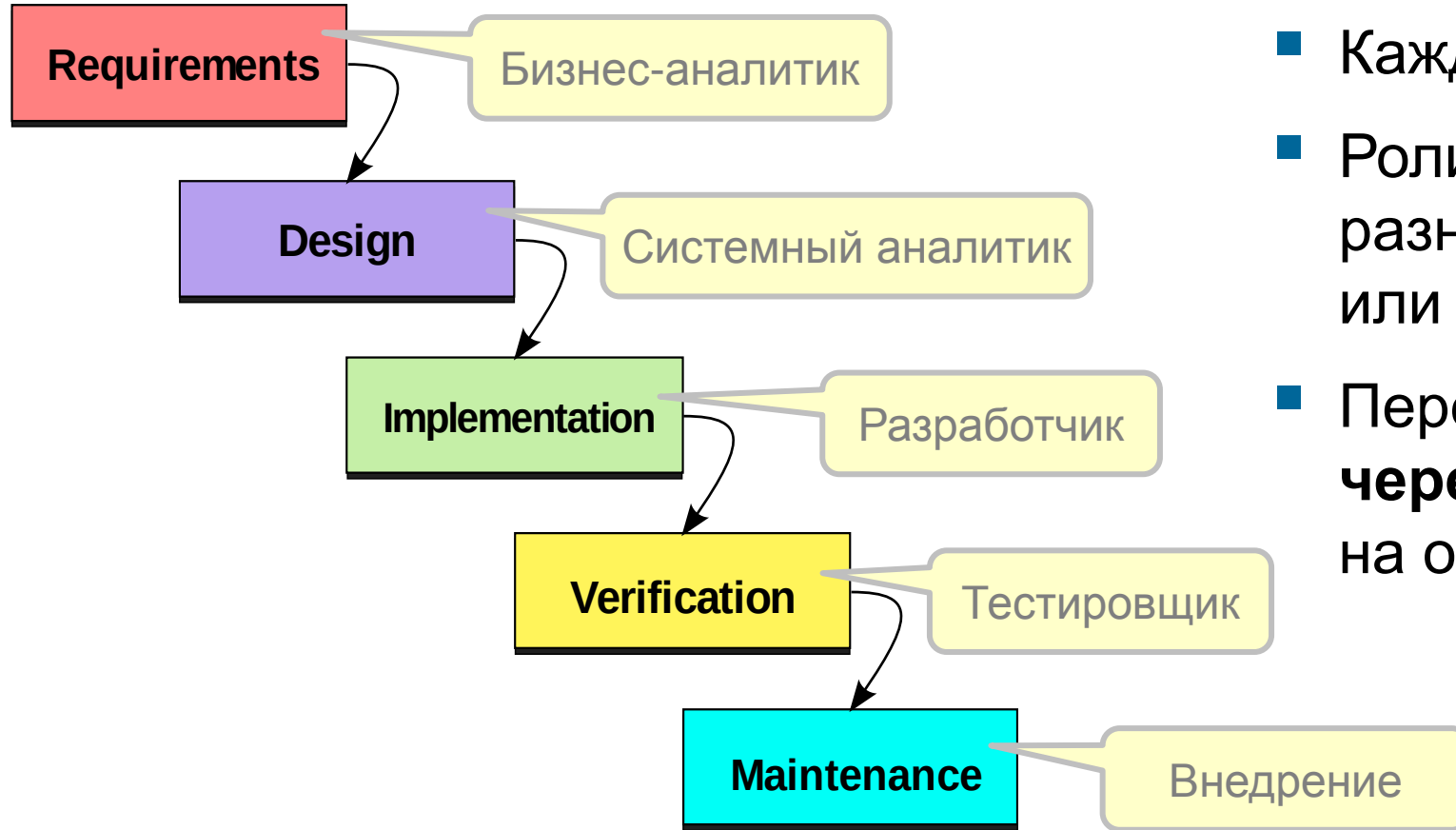
Минск, 07 февраля 2020

Правильного разделения нет



Но для многих *естественно*
думать иначе

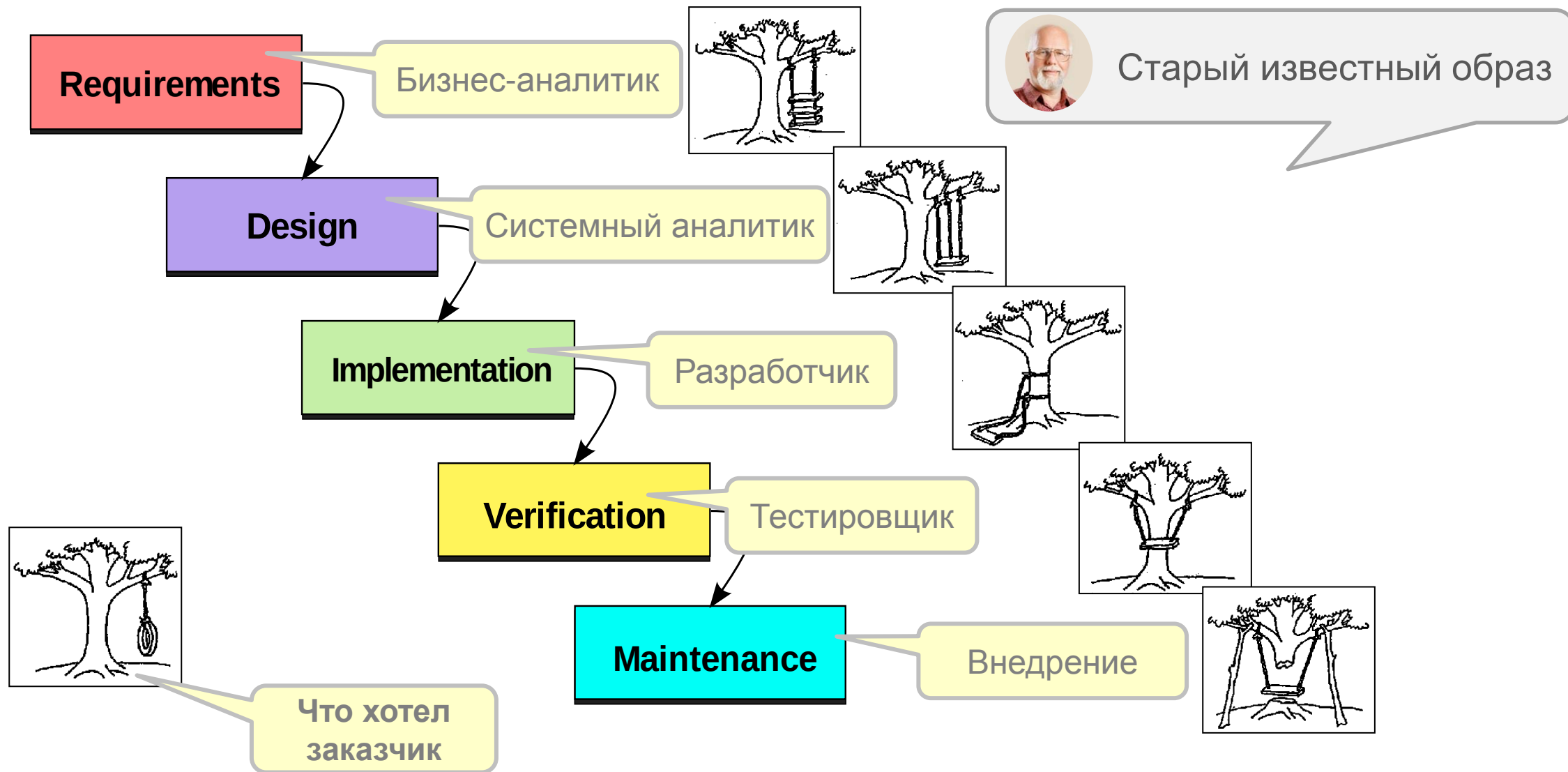
Водопад – специализация по фазам проекта



- Каждой фазе – своя роль
- Роли выполняются разными людьми или командами.
- Передача работы – **через артефакты** на отдельных языках.

Модель водопада – http://en.wikipedia.org/wiki/Waterfall_model

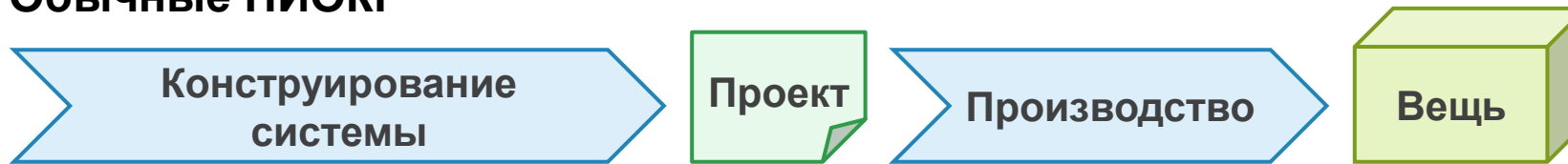
Каждая передача искажает смысл...



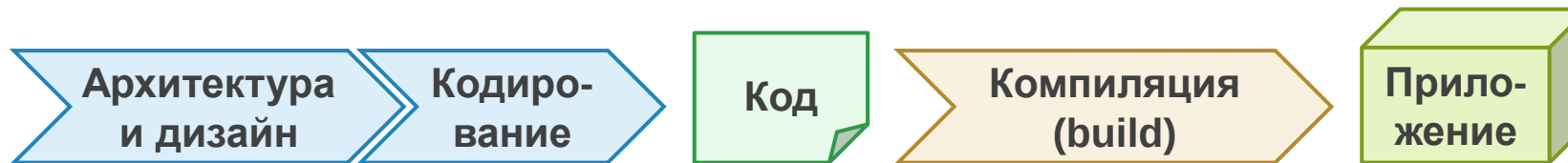
Почему плохо работает?

[Jack W. Reeves. What is software design](#), 1992 ([перевод](#))

Обычные НИОКР



IT-разработка



Мешает природа IT-разработки:
разработка кода – часть проектирования

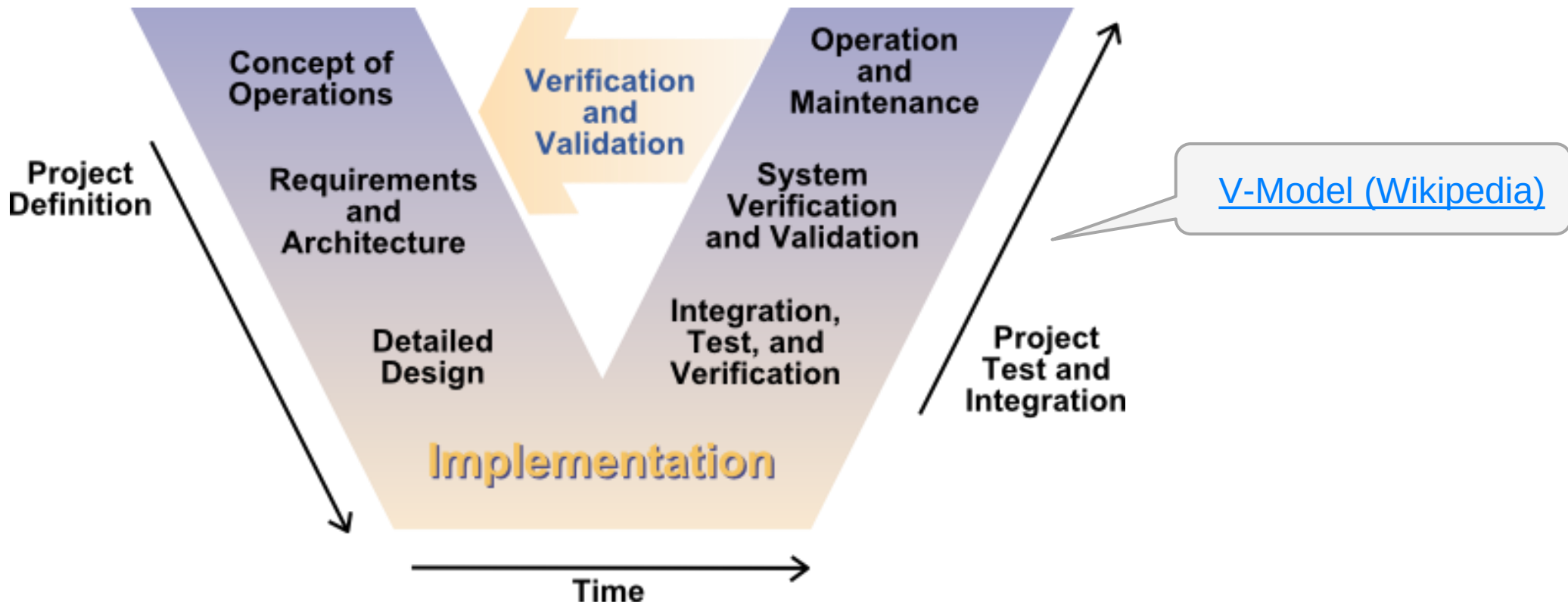
Agile – ответ на провал процедур

- Успех проекта определяют люди – это было осознано еще в 1980-х
 - Фредерик Брукс «Мифический человеко-месяц», 1975)
 - Том ДеМарко «Человеческий фактор» (1987)
- Этому не поверили и в конце 1990-х был поставлен эксперимент по нормированию процессов – RUP. Он окончился неудачей
- Появление персоналок кратно усилило потребность в разработке
- Agile появился в 2000-х, Scrum обеспечил успех
 - [Джеф Сазерленд на SECR-2011](#) Scrum как высшая стадия CMMI ☺
 - [Dan Rawsthorne на AgileDays-2012](#) ([слайды](#)) Scrum: the Big Picture

Подробнее – мой доклад «[Мыслить проектно](#)» на SECR-2018

Agile объединяет роли
и проводит новые границы

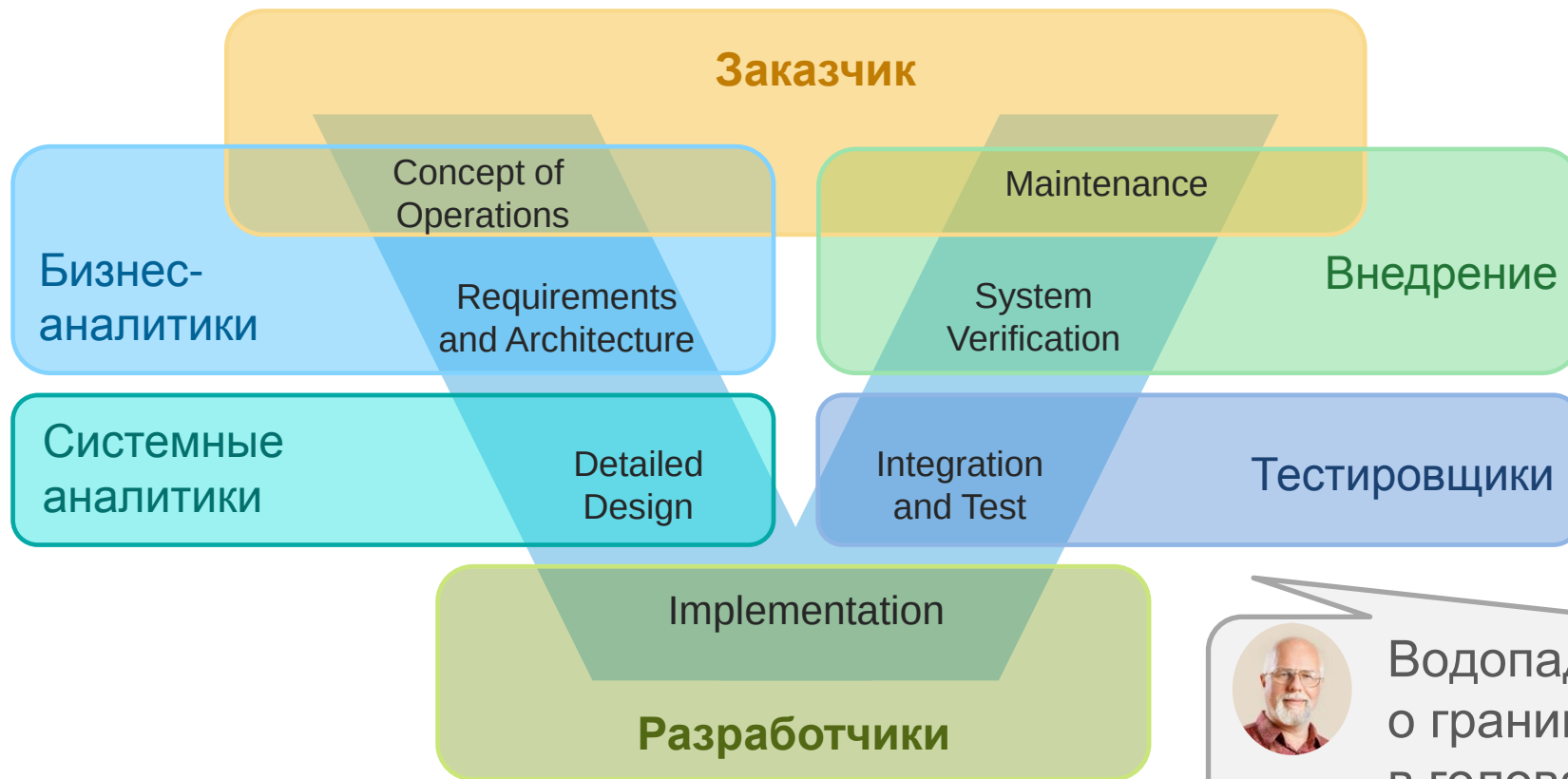
Визуализация – V-модель



Чтобы проводить границы нужно непрерывное, а не дискретное представление

Специализации из водопада на V-модели

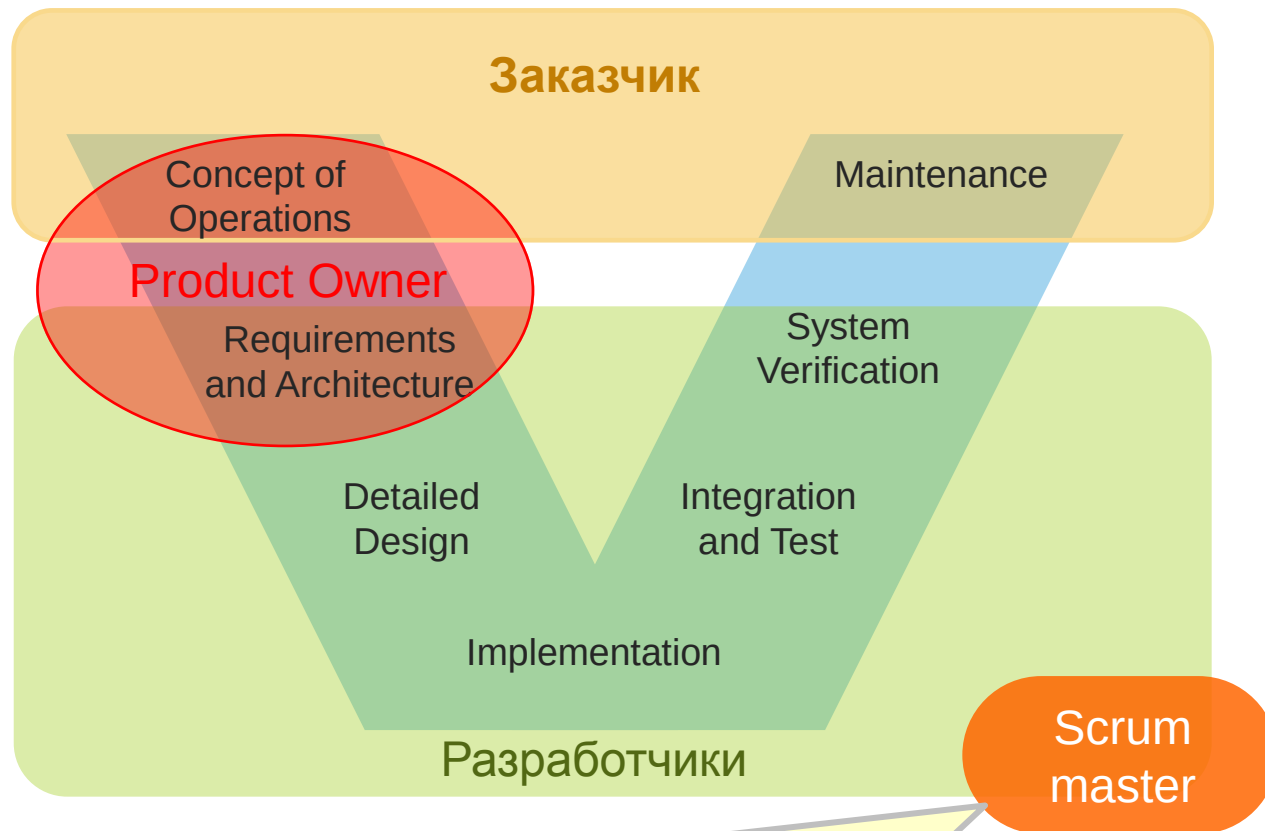
- ☹ Коммуникации – лишь с соседями.
- ☹ Длинный цикл общения ведет к потере информации.



Водопад ушел, а представления о границах ответственности в головах и учебниках остались

В команде – только разработчики

- Кросс-функциональная команда разработчиков.
- Product Owner взаимодействует с заказчиком

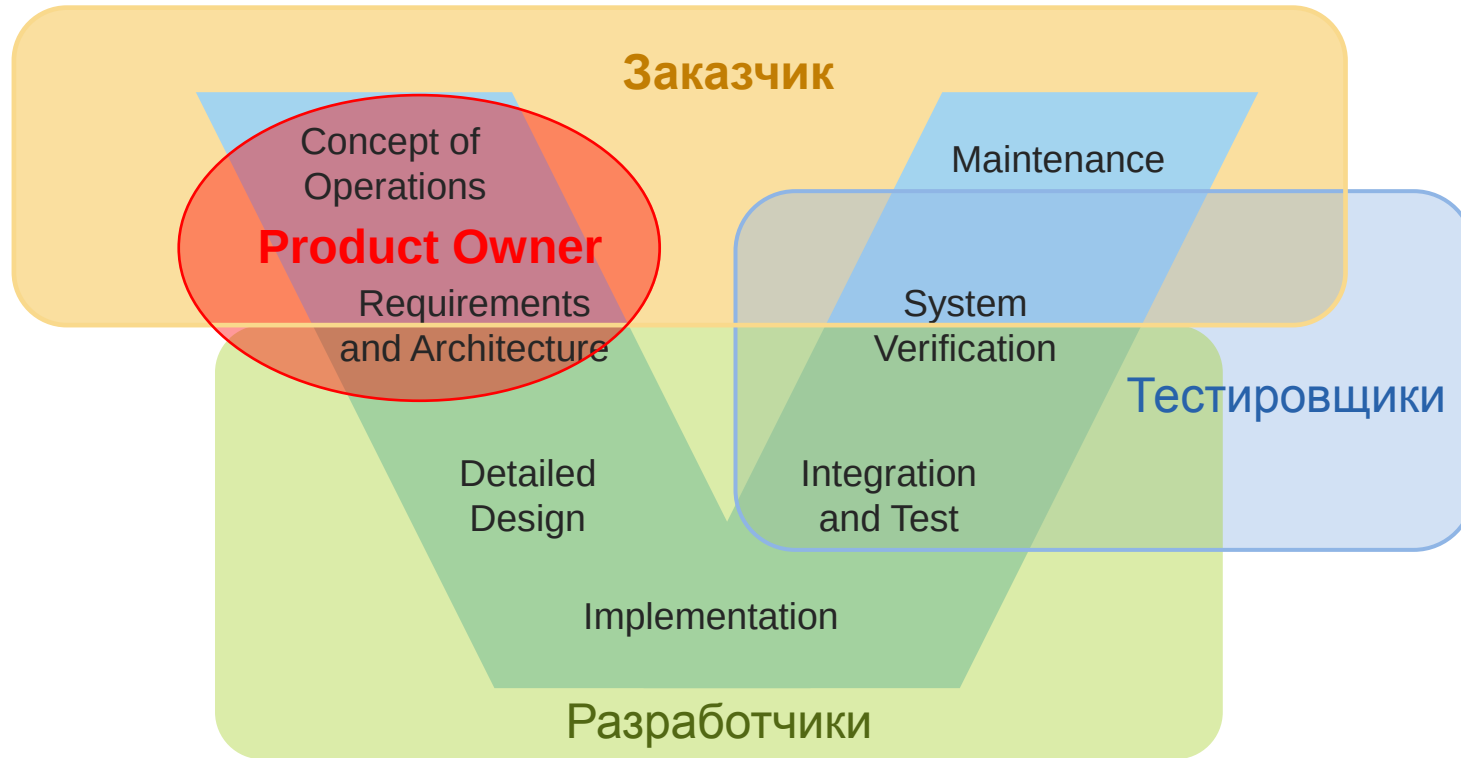


- 😊 Эффективные коммуникации
- 😊 Возможность быстрой обратной связи
- 😞 Большая нагрузка на Product Owner'а
- 😞 Слишком разнообразная работа членов команды

Схема SCRUM, в других методах – аналогично

Возвращение специализаций

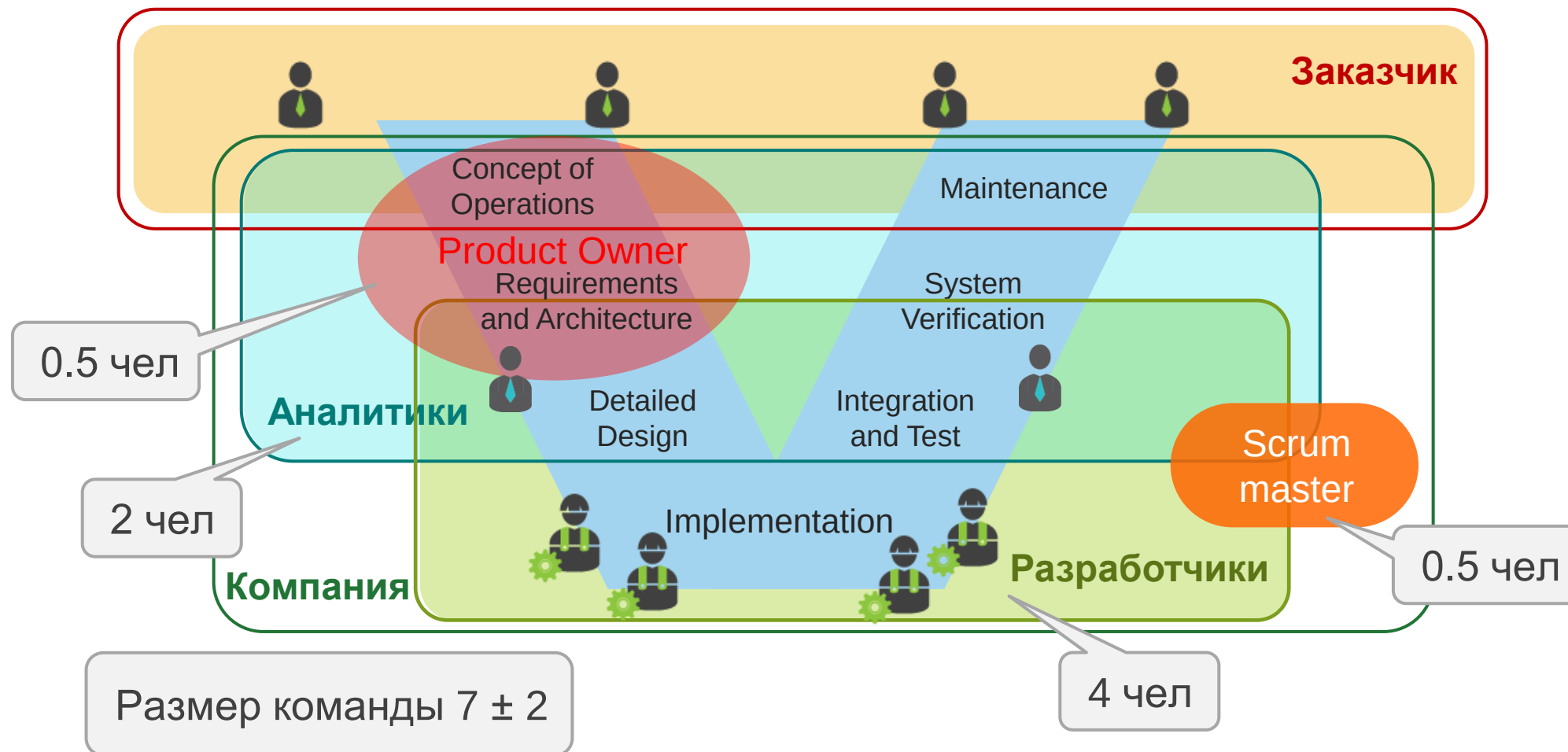
Команда с тестировщиками



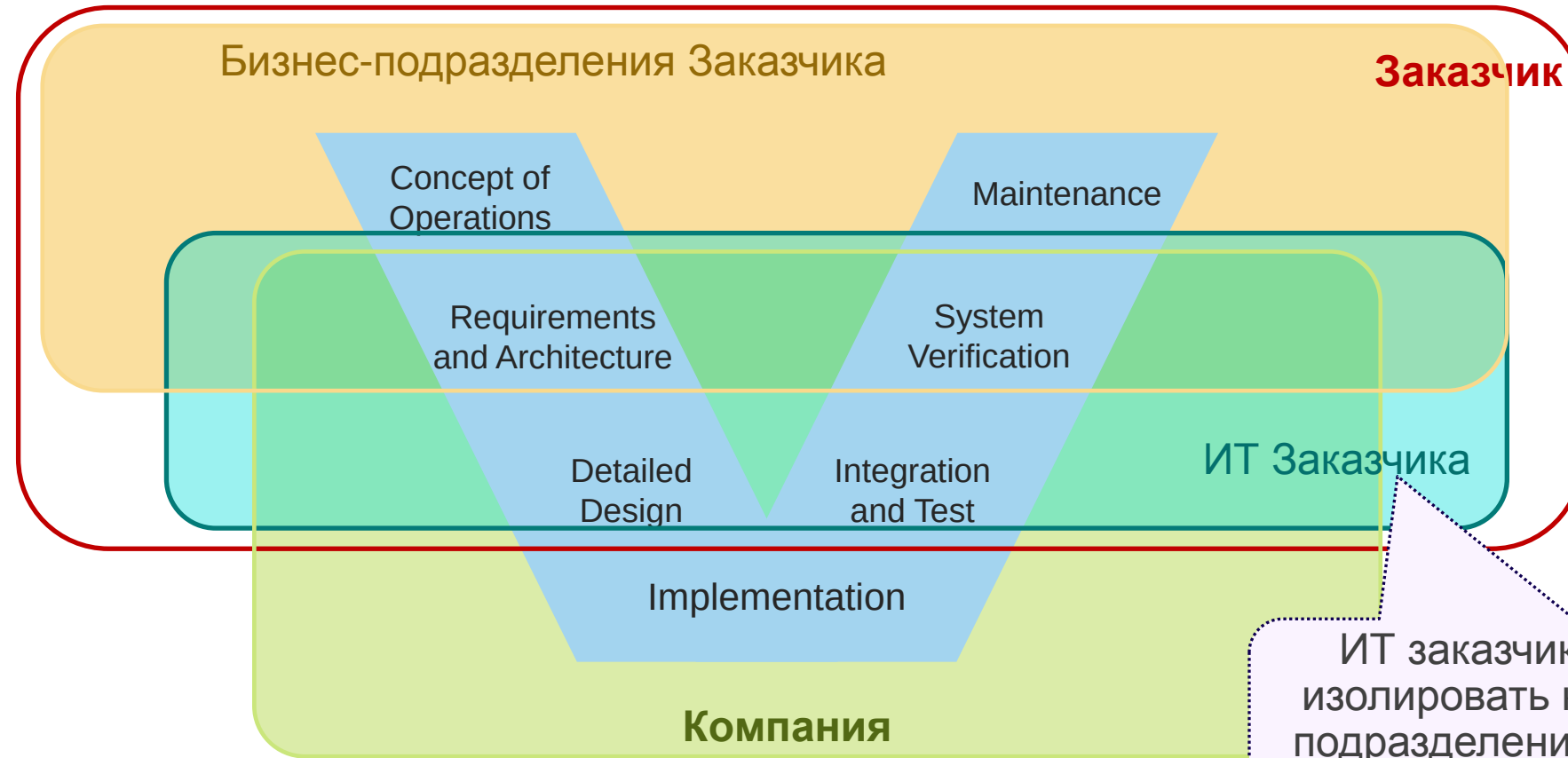
Складывается, когда заказчик готов формулировать требования

Аналитики между заказчиком и разработкой

Формулируют бизнес-задачу, проектируют организационную часть и требования к ИТ-системе, а затем – принимают и внедряют результат

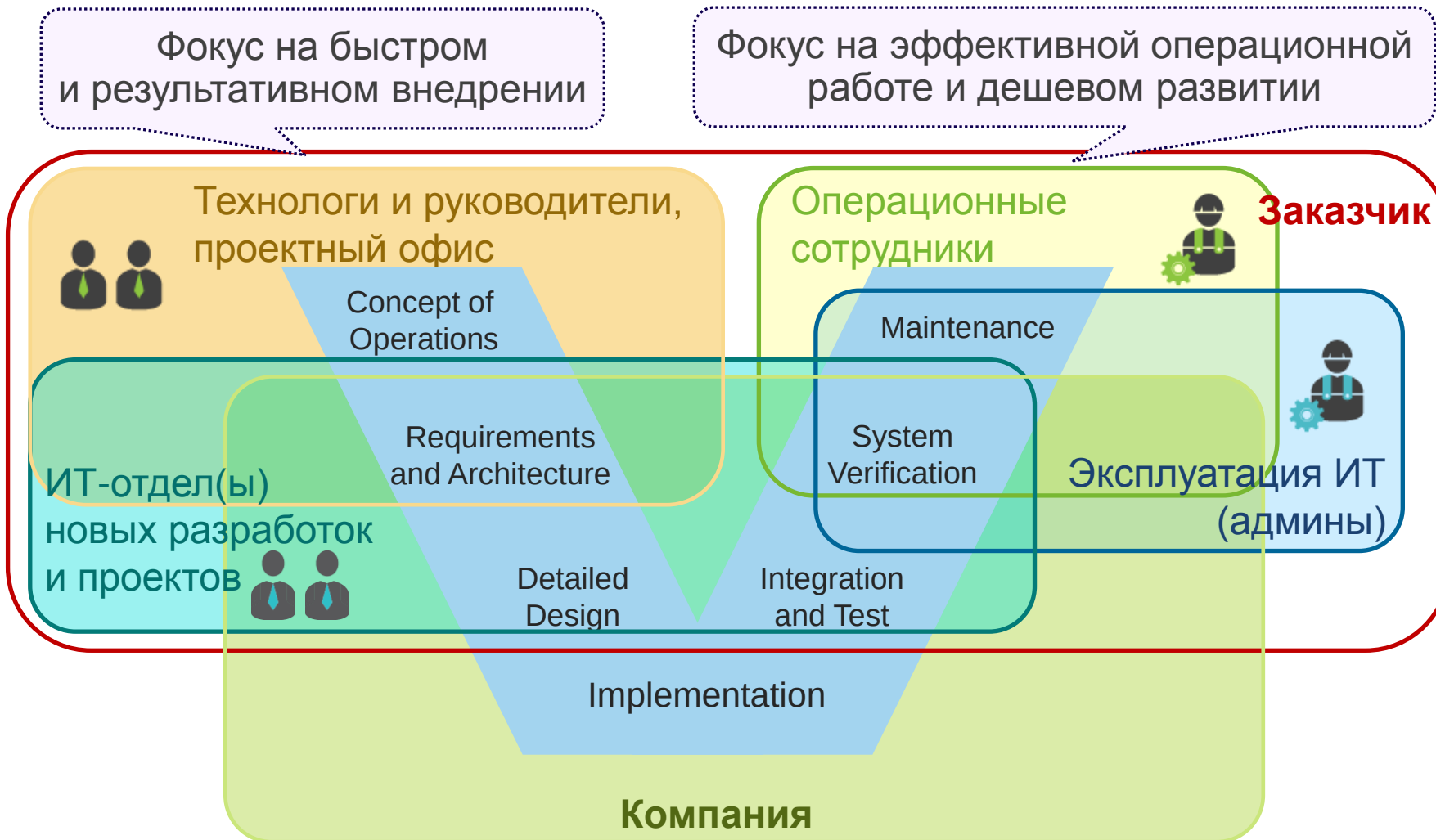


У заказчика есть свой ИТ...



ИТ заказчика часто стремится изолировать компанию от бизнес-подразделений, однако для успеха проекта взаимодействие необходимо

Стейкхолдеры разработки и эксплуатации



Аустсорсинг кодирования

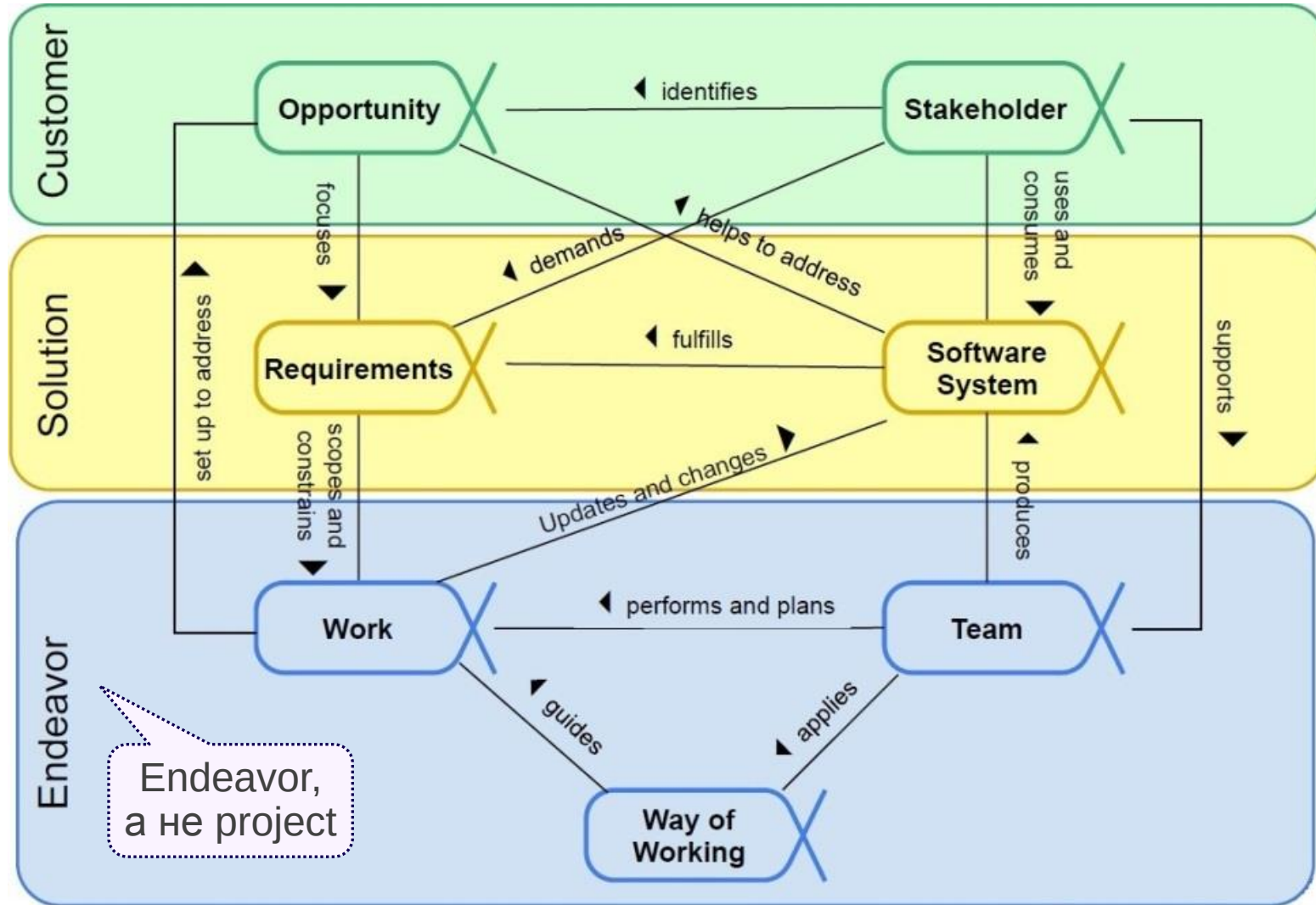


Где проходит граница с Заказчиком у вас?

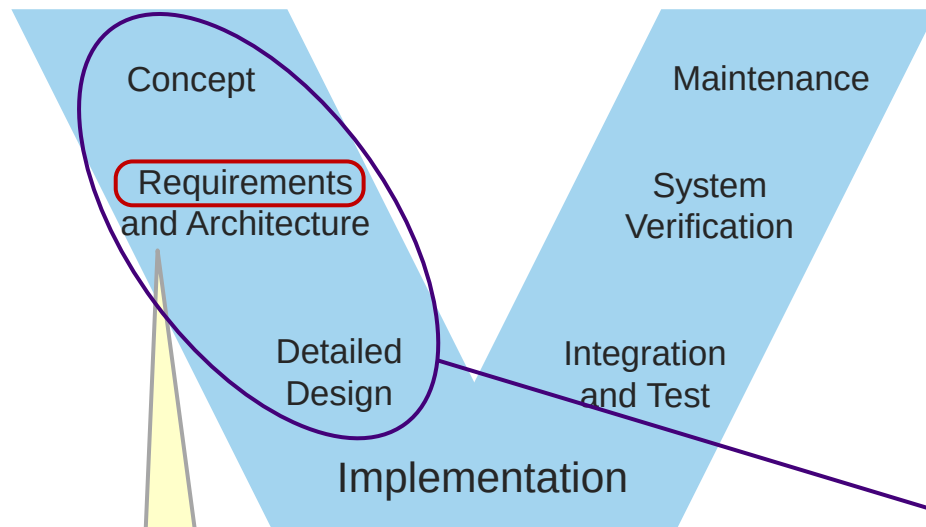
- Заказчик рассказывает о проблемах и потребностях, или выдает техническое решение, требуя ему следовать?
- Если оказывается, что решение (после реализации) не устраняет проблему, рассчитывает ли Заказчик на бесплатное исправление?
- Есть ли доступ к конечным пользователям решения, или к их бизнес-руководителям, или контакт – IT-специалисты?
- Ожидается ли в документах описание бизнес-проблем?
- Проверяет ли бизнес Заказчика технические решения?

Многофокусное ведение проектов

Альфы OMG Essence – фокусы внимания

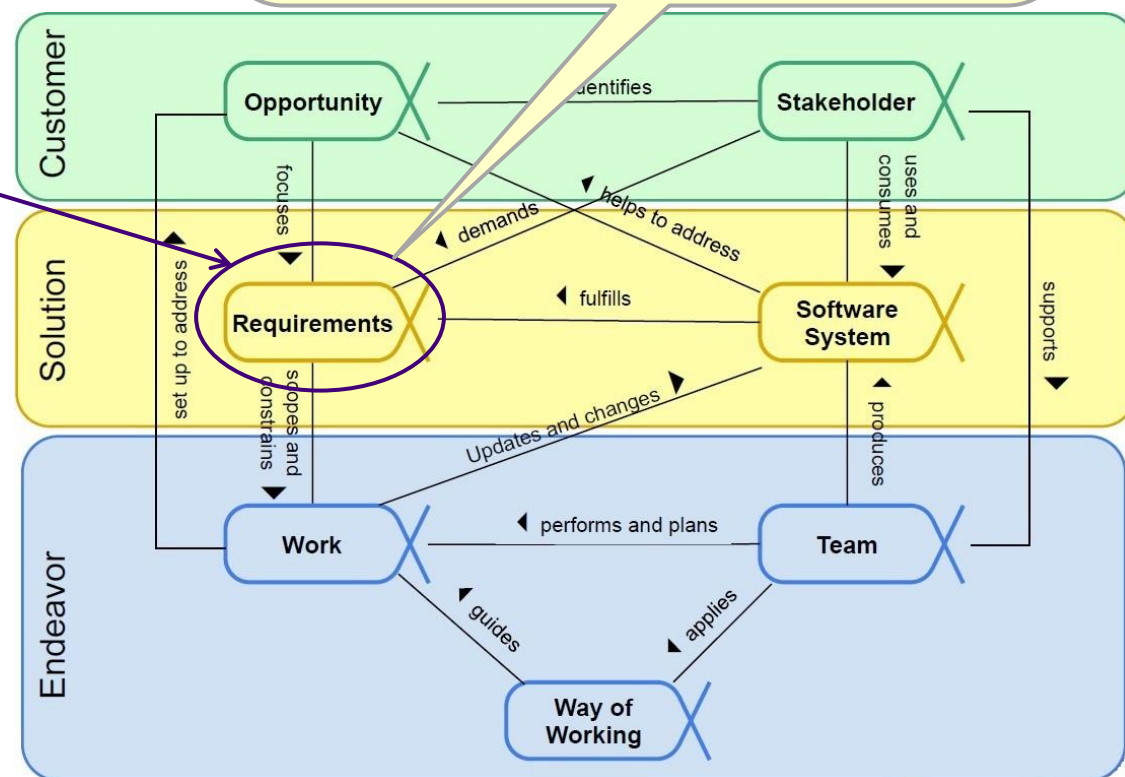


Что такое требования – два ответа



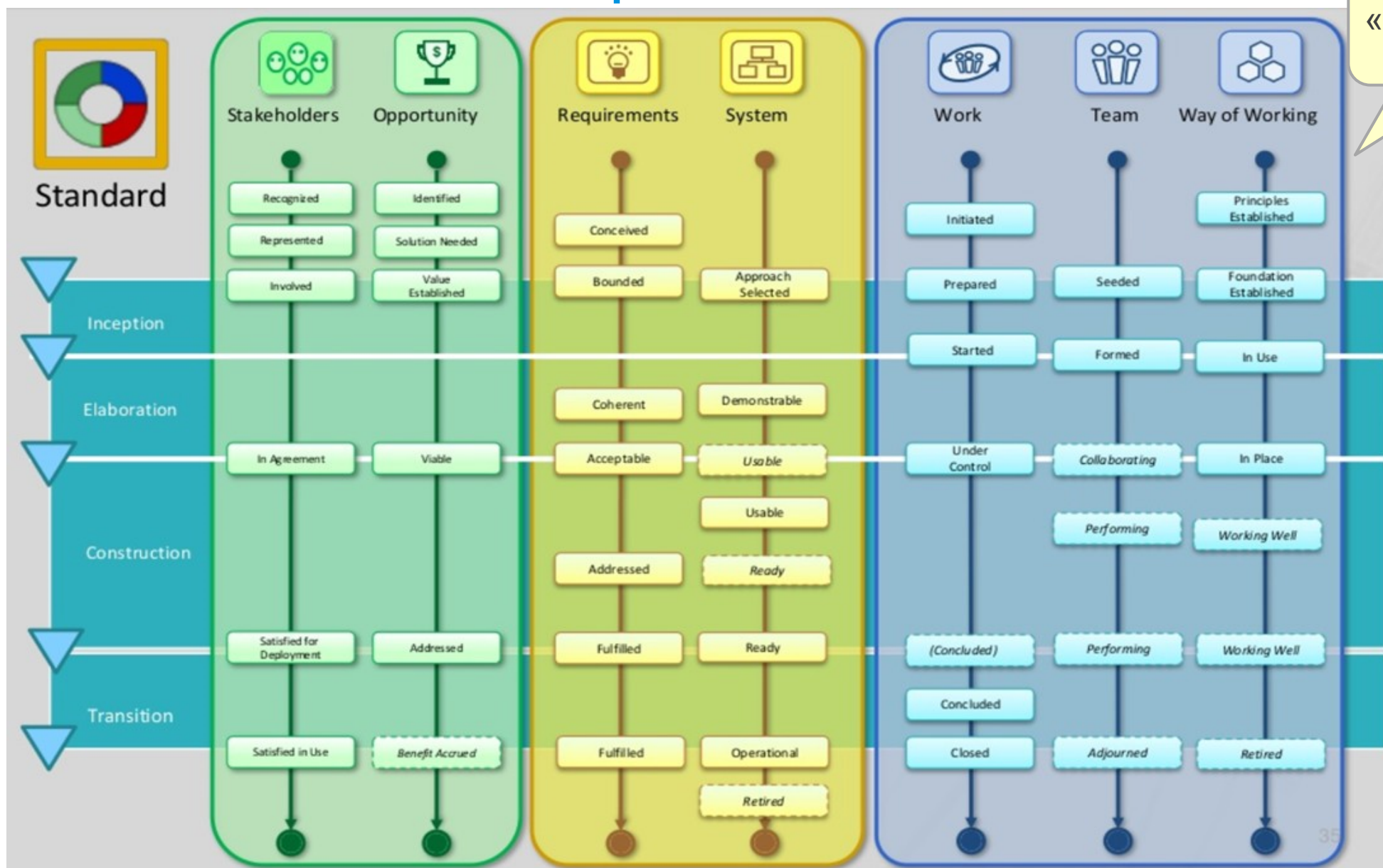
На V-диаграмме ([Wikipedia](https://en.wikipedia.org/wiki/V-model))
требования – внешние функции системы

В [OMG Essence](https://www.omg.org/essence) требования –
все описание системы
от потребностей пользователей
до детального дизайна.
Возможности – отдельно



Жизненный цикл проекта в OMG Essence

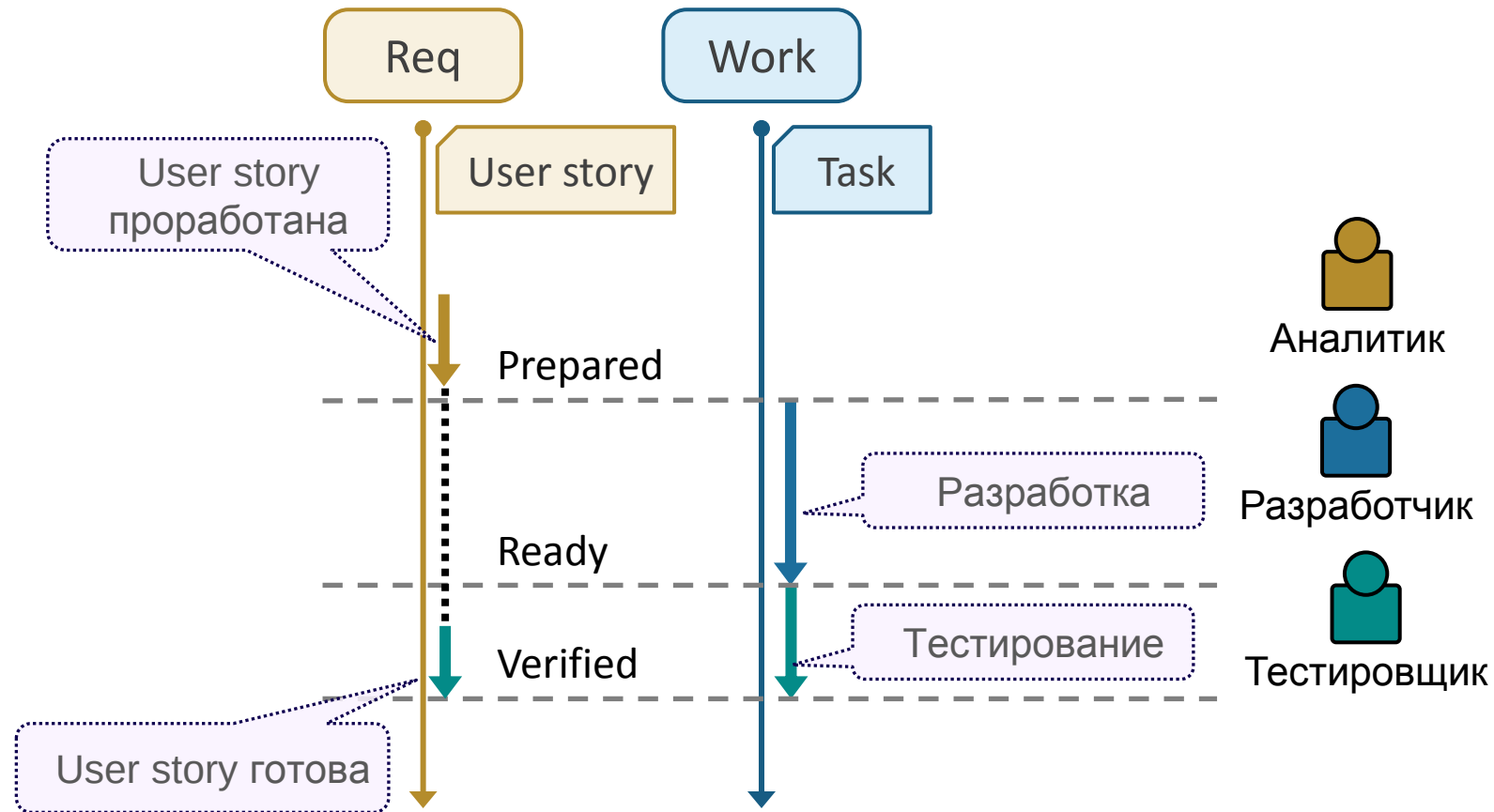
Пример:
«стандартного»
проект



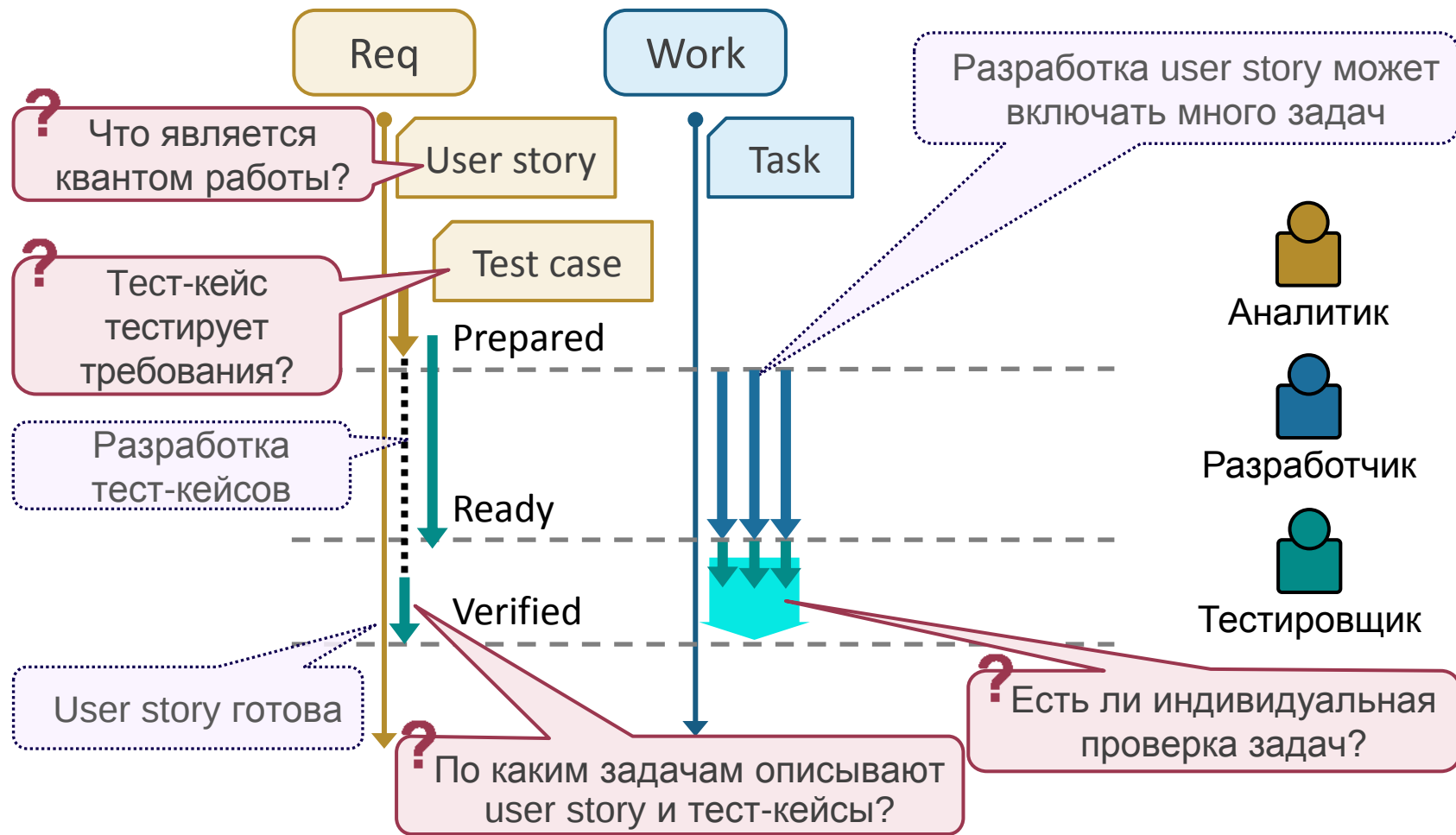
Использование Lifecycle OMG Essence

- Альфы – абстрактные объекты, состояния которых отражают движение проекта
- Альфы Представлены артефактами – физическими объектами
- Ответственность делится по этим объектам
- Check lists состояний определяют, в чем ответственность, в стандарте есть стартовый готовый набор
- Lifecycle-диаграмма иерархична: можно представить весь проект, его релиз, спринт или разработку фичи

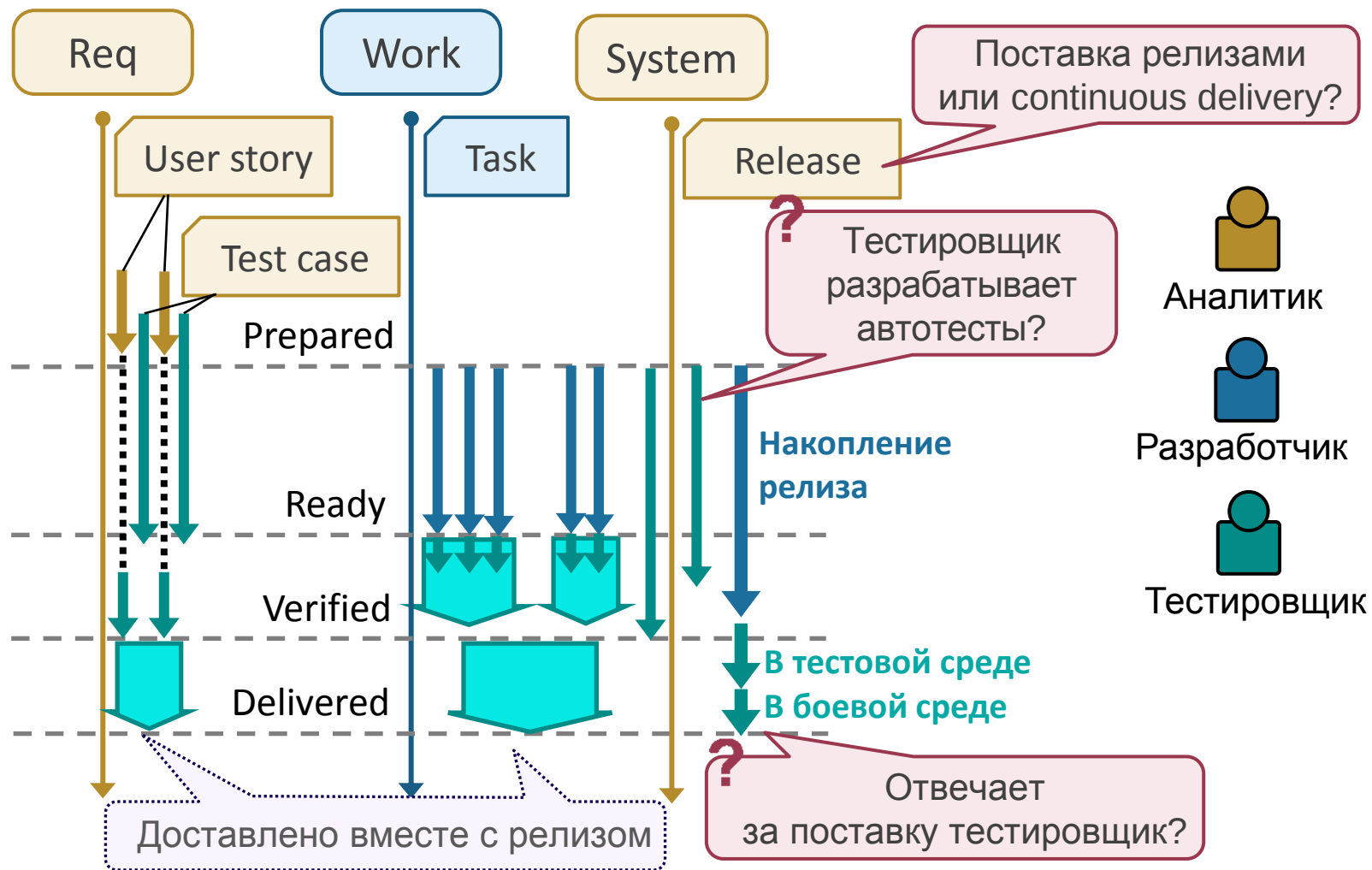
Простейший вариант – ответственность за задачу



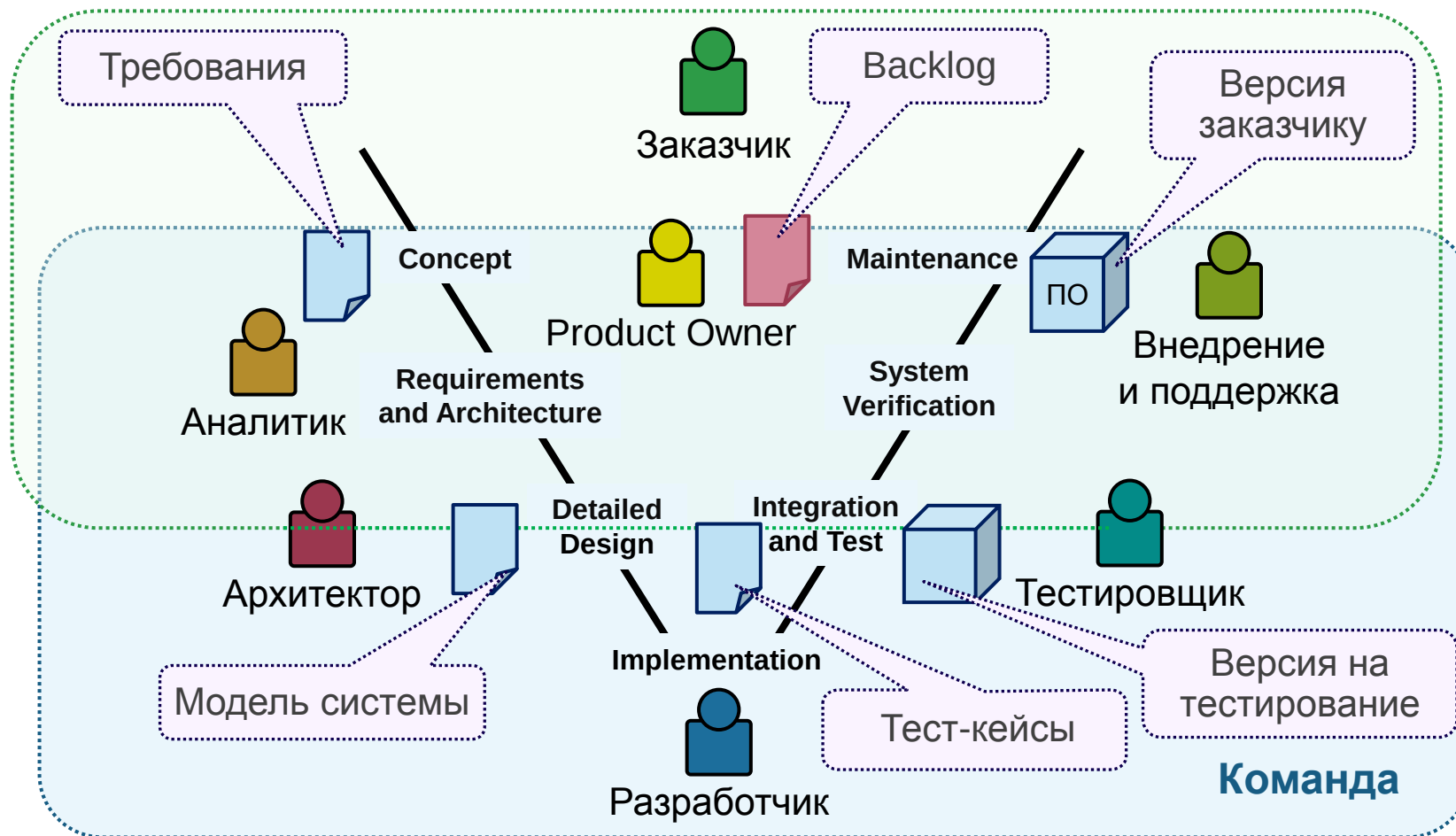
Ответственность за готовность user story



Ответственность за поставку системы



Ответственность за артефакты



Разные границы проекта

Смена культур ИТ-проектов

Рамка проекта:

ИТ-система...

... обеспечивает
бизнес

... делает то,
что нужно

... сделана
вовремя

... работает

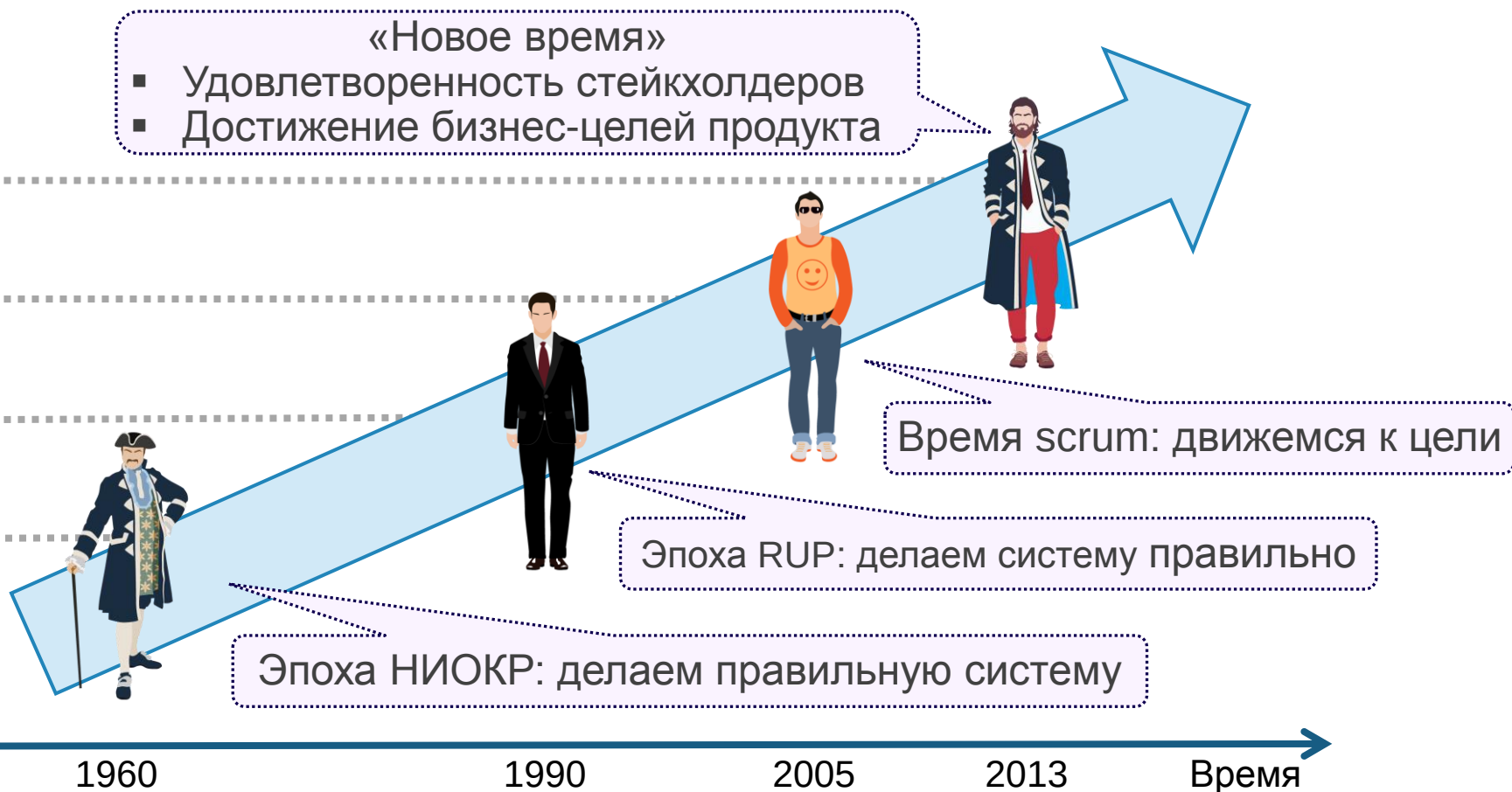
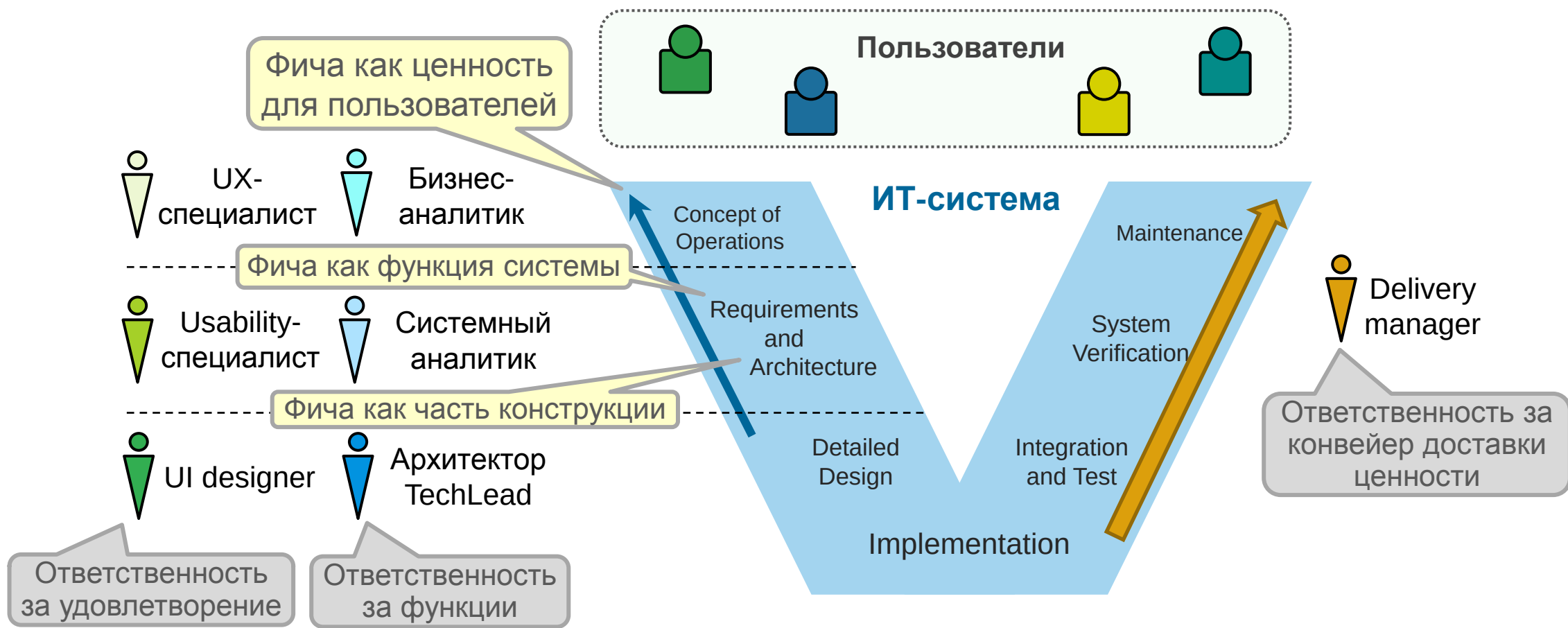


Схема из моего доклада «[Мыслить проектно: история и современность](#)» на SECR-2018, впервые рассказана [на AgileDays-2015](#)

Вперед к пользователям!

Ориентация на пользователей в Public web
породила отдельную линейку специализаций



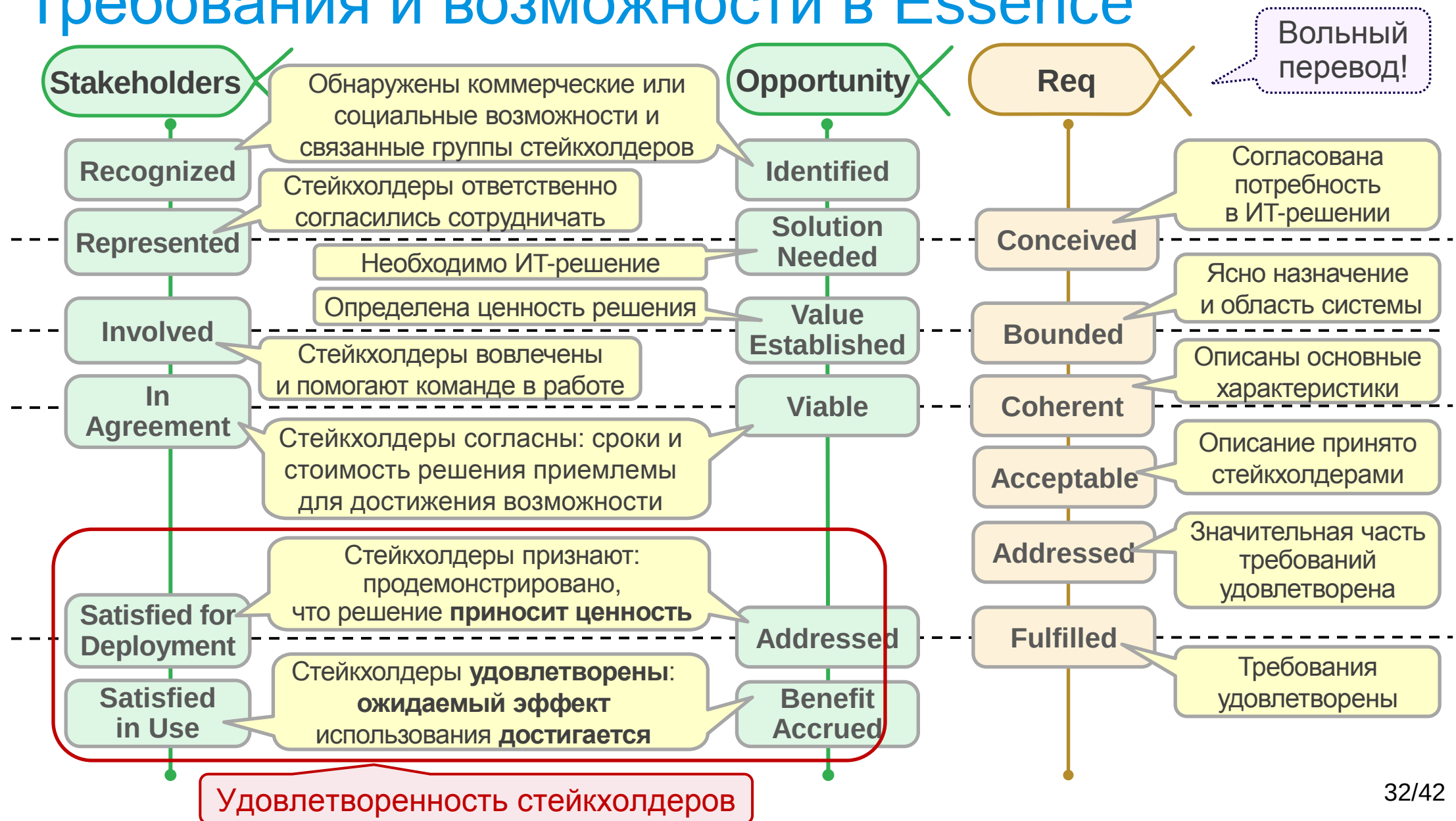
Карта специализаций

- Разработчики специализируются
 - по технологиям: Java и СУБД, web-дизайн и разработка и т.д.
 - по квалификации: выделяется архитектор, техлид
- Другие роли тоже специализируются
- Skills Framework for Information Age sfia-online.org
 - 97 компетенций в 6 категориях и 17 подкатегориях
 - 7 сквозных уровней, характеризующихся по autonomy, influence, business skill: follow, assist, apply, enable, advise, initiate, strategy
 - Образование договаривается с бизнесом
 - Использование: professional skill + behavioural skill + knowledge + experience + qualifications
- SFIA позволяет сформулировать требования к дополняющим позициям в проекте

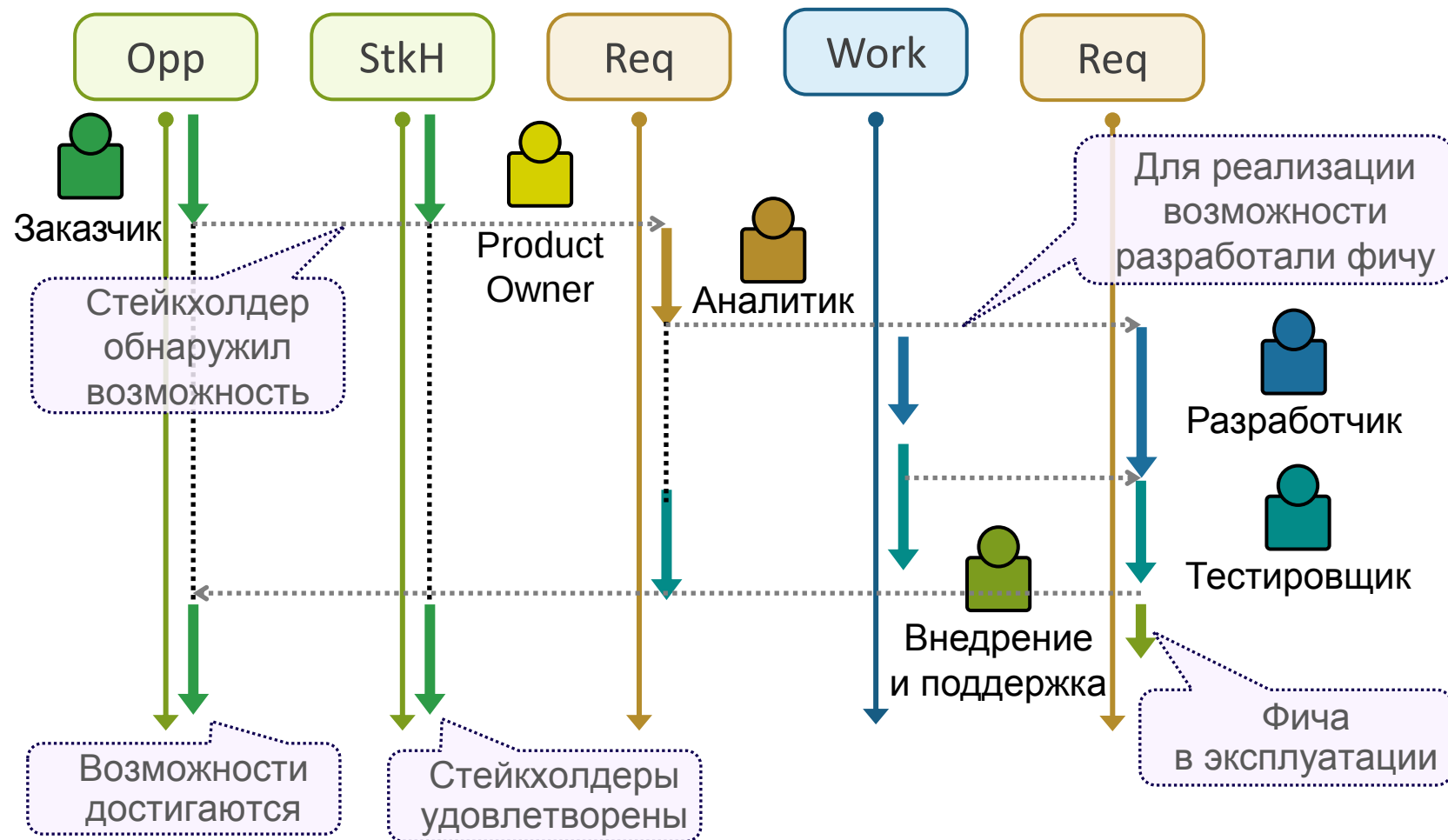
ИТ- и бизнес-проект объединяются



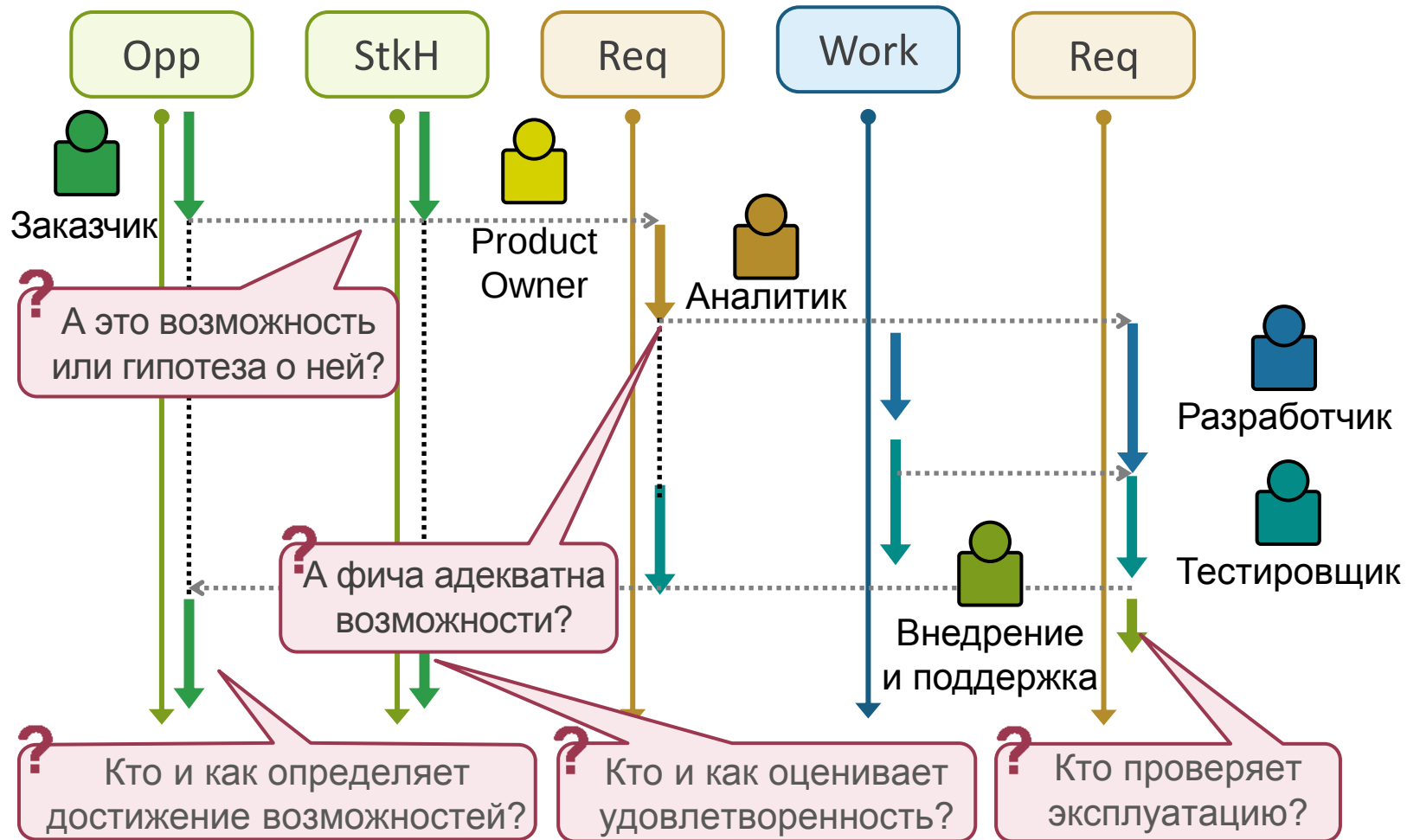
Требования и возможности в Essence



Удовлетворенность стейкхолдеров и обеспечение возможностей бизнеса



Кто проверяет цели и их достижение?

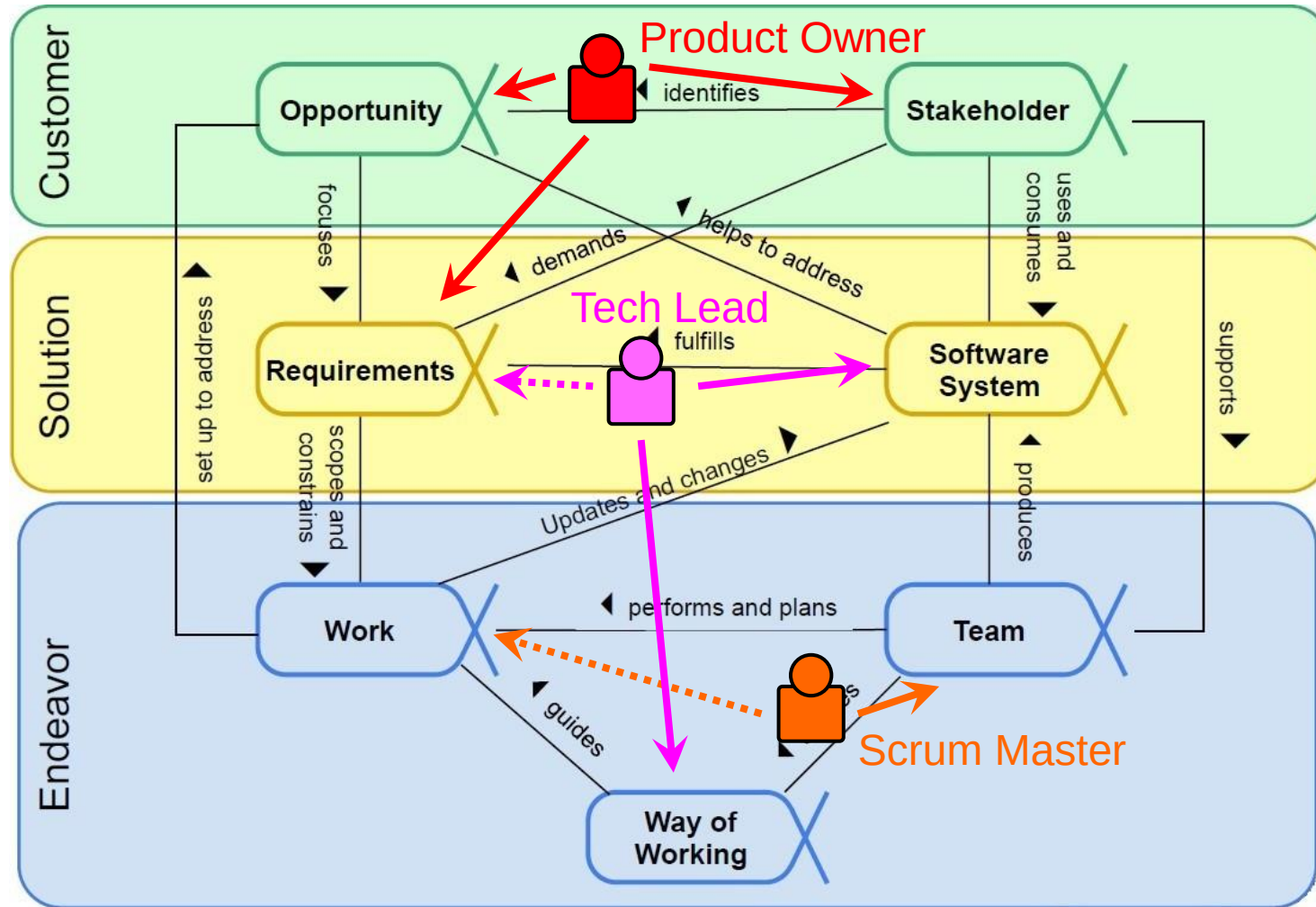


Ответственность за управление

Ответственность за управление

- Классический подход: за все отвечает менеджер
- Agile разделяет ответственность
 - Product Owner отвечает за продукт
 - Команда самоорганизуется для выполнения работ
 - Scrum master помогает эффективной самоорганизации
 - Менеджер тоже есть, но на следующем уровне управления
- Если квалификации недостаточно – ветка техлидов, отвечающих за технологии разработки

Ответственность за управление



Управленческая ответственность

Егор Толстой, Стас Цыганов и сообщество

[GitHub/TLbootcamp/TLroadmap](https://github.com/TLbootcamp/TLroadmap)

- People manager
- Product Owner
- Integrator
- Technical lead
- Administrator

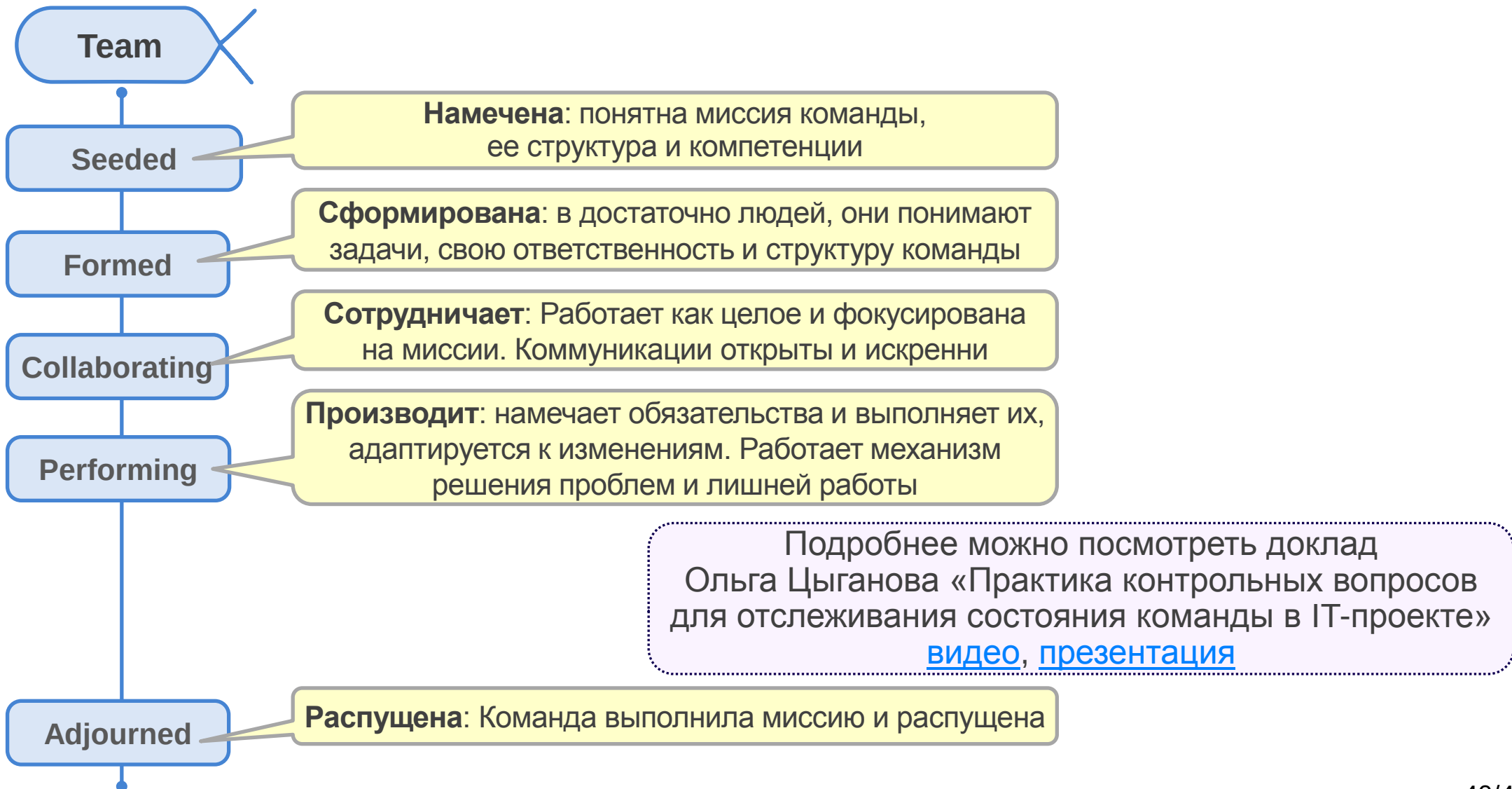


Personal Skill for Team Lead

Егор Толстой, Стас Цыганов и сообщество
[GitHub/TLbootcamp/TLroadmap](https://github.com/TLbootcamp/TLroadmap)



Альфа команда



Заключение

- Не спорьте о том, какой путь «самый верный»
- Создавайте разделение ролей исходя из проекта
- Для визуализации хорошо использовать V-модель
- Если надо детально – используйте фазы и чек-листы OMG Essence
- Эффективные коммуникации – необходимы
- Разделение ролей определяет компетенции
- Учитывайте какие специалисты есть на рынке



Вопросы и кейсы – обращайтесь!

Максим Цепков

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

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