Роли в проекте: как поделить поляну ответственности

Максим Цепков
http://mtsepkov.org

Точка Сборки
конференция Санкт-Петербургского сообщества аналитиков
09 сентября 2017

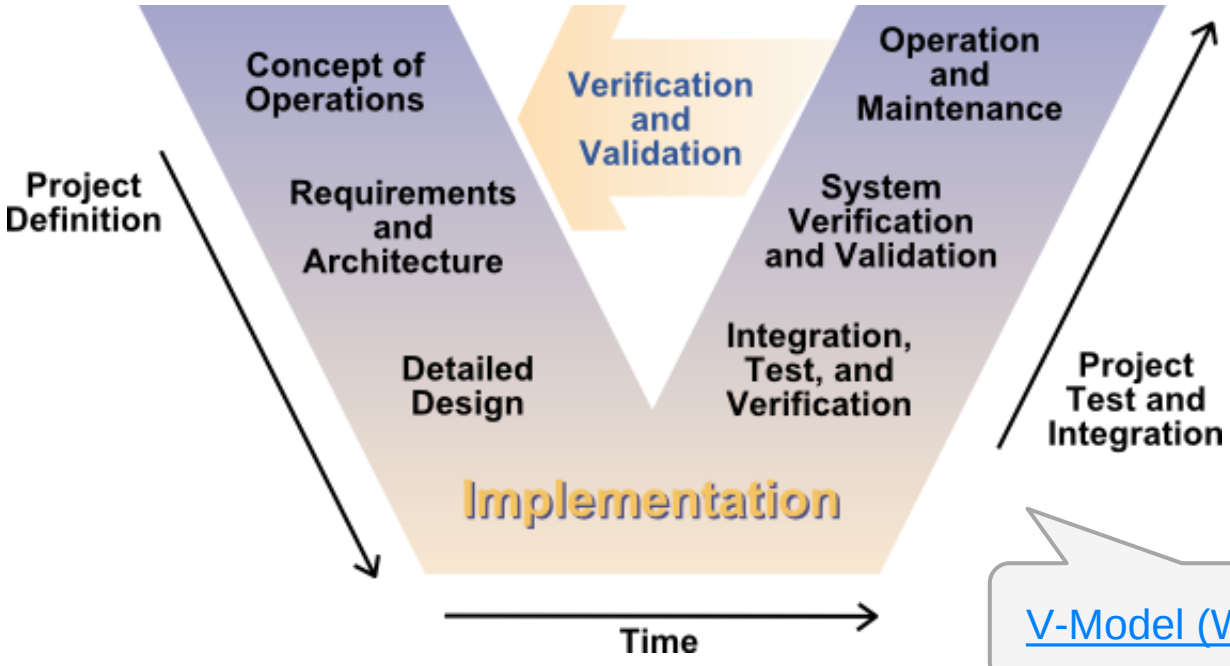
О чем будет доклад

- Модель процесса разработки изменялась
  - Классика – водопад и RUP, специализации пришли оттуда
  - Agile качнул модель до полной кроссфункциональности
  - Потом специализации вернулись
  - **OMG Essence** – **конструируем** способ ведения проекта
- Я дам обзор вариантов для разных проектов
  - Специализации внутри команды
  - Разные границы проекта
  - Разделение ответственности между командами

Визуальное представление

Диаграммы – **фишка**

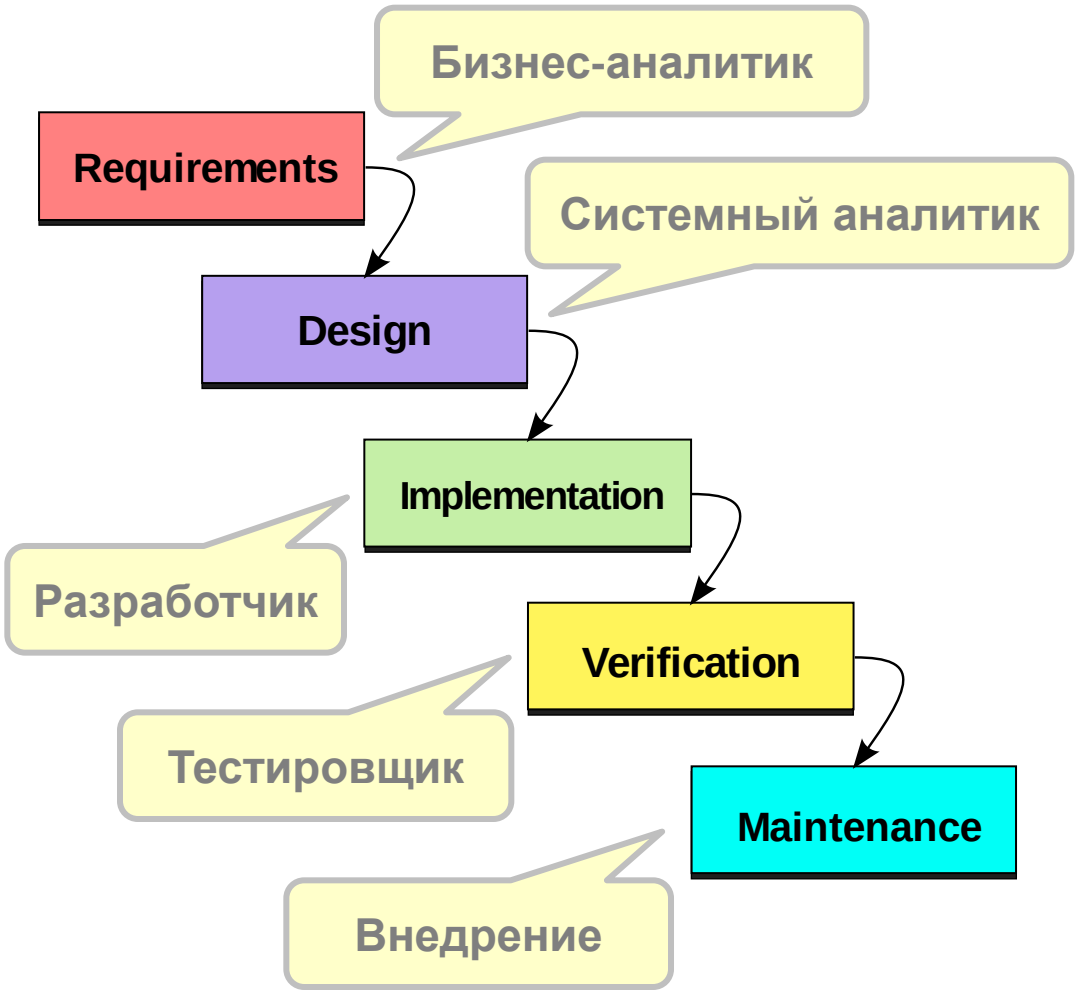
- Возьмем схему процесса – и разметим
- Используем стандартную схему – V-модель



V-Model (Wikipedia)

Делим поляну внутри команды

Водопад – специализация по фазам

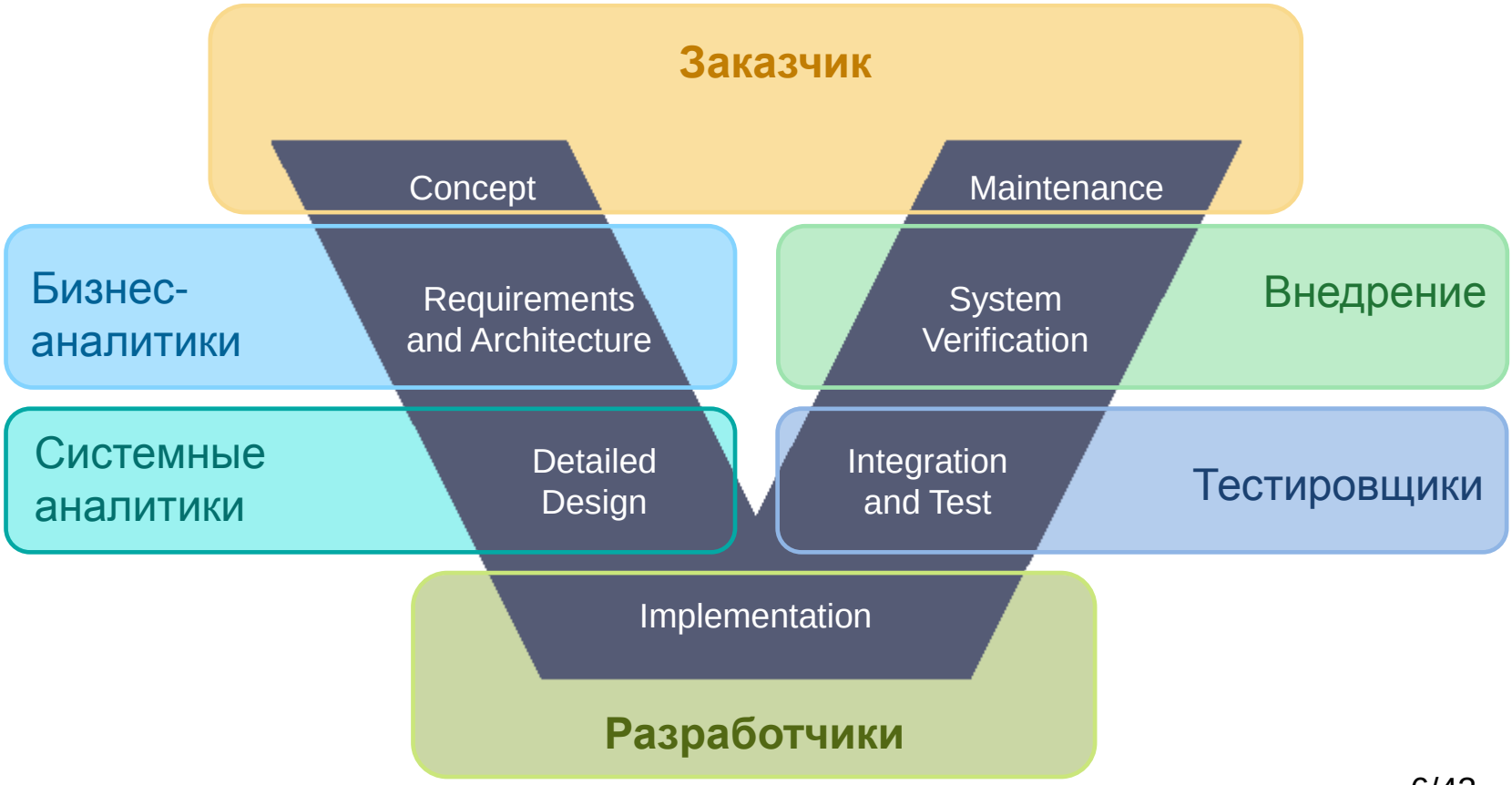


- Каждой фазе – своя роль.
- Роли выполняются разными людьми или командами.
- Передача работы – **через артефакты** на отдельных языках.

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

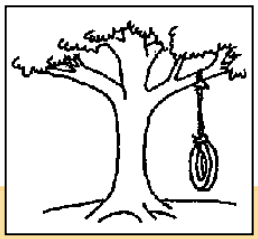
Водопадная модель на схеме

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

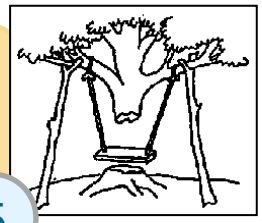


Изменение видения проекта…

Старый известный образ



Что хотел заказчик



Заказчик

Concept

Maintenance

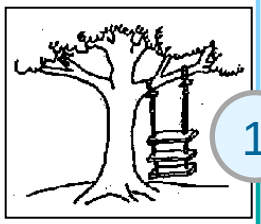
5

Бизнес-аналитики

Requirements and Architecture

System Verification

Внедрение



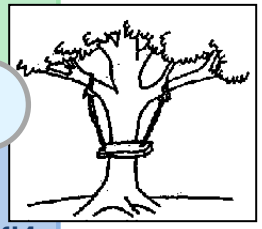
1

Системные аналитики

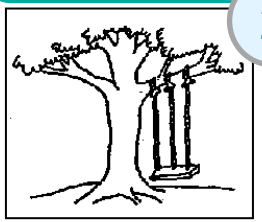
Detailed Design

Integration and Test

4



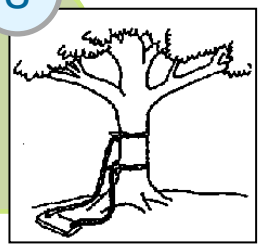
Тестировщики



2

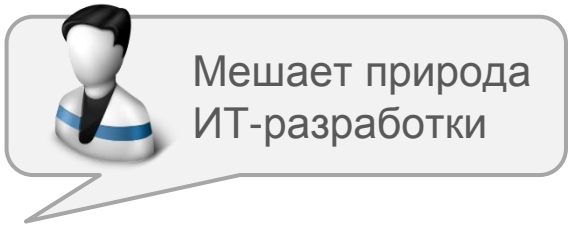
3

Implementation



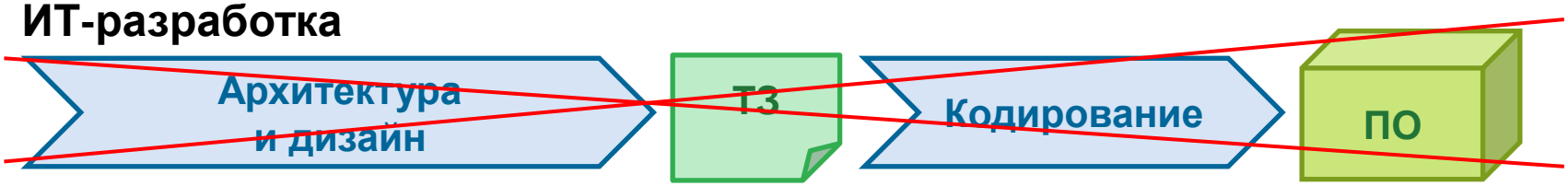
Разработчики

Почему плохо работает?



Обычный НИОКР

ИТ-разработка

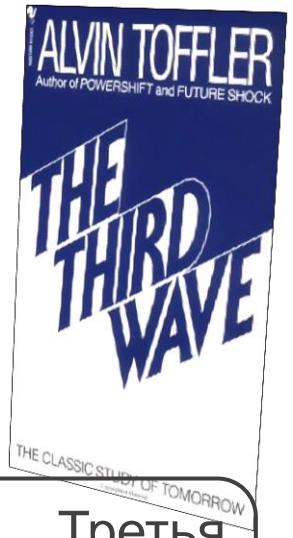


Jack W. Reeves «What is software design» (1992; перевод)

Agile – ответ на провал процедур

- Успех проекта определяют люди – это было осознано еще в 1980-х, и обосновано **Том ДеМарко** в книге «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
- А сейчас Agile идет в другие отрасли

Agile и традиционный менеджмент



Agile возник как альтернатива классике

Основной путь – Agile вбирает традиционные подходы и заменяет классический менеджмент

Agile-менеджмент

Третья волна Тоффлера

Традиционный менеджмент

Вторая волна Тоффлера

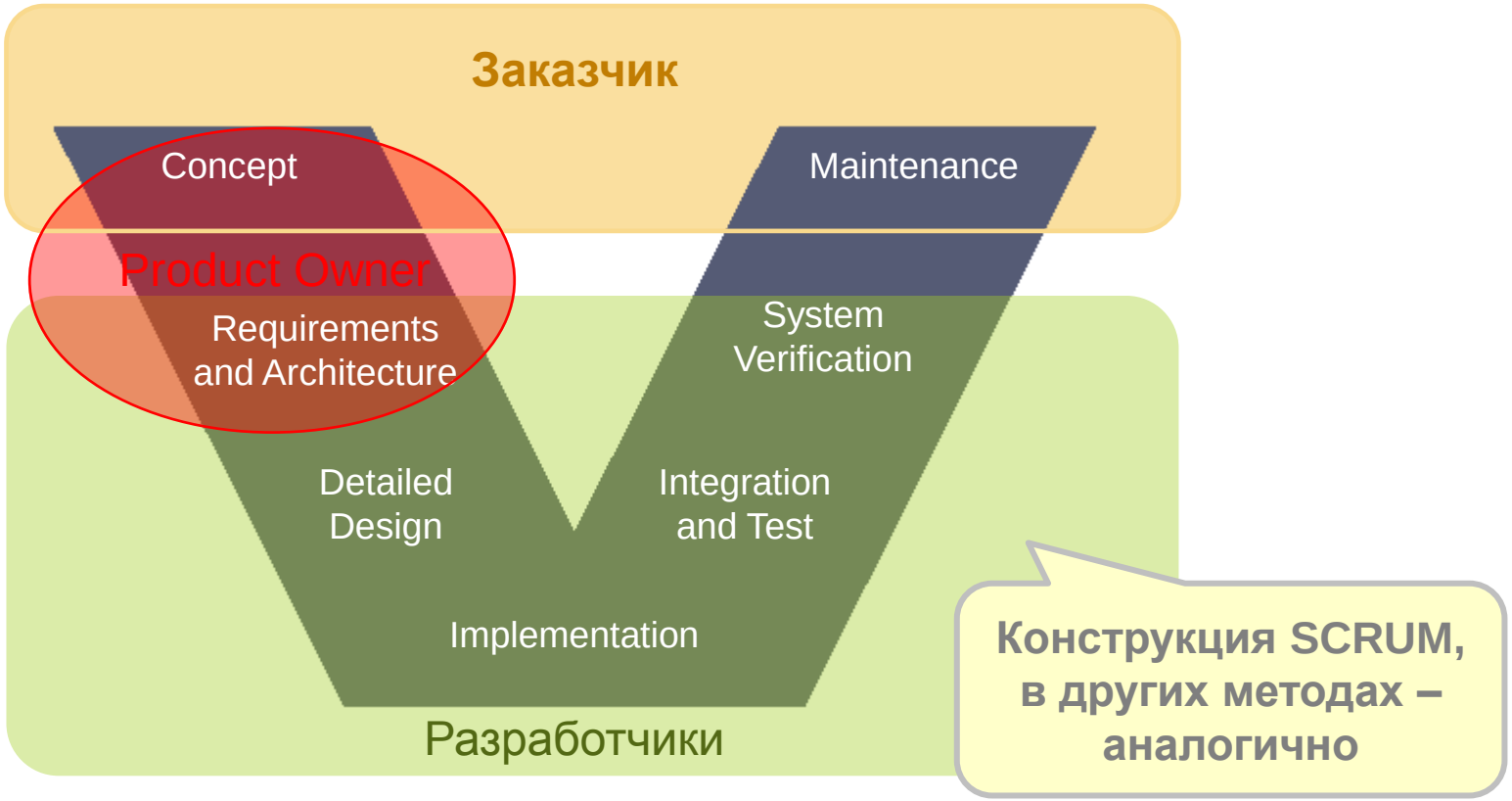
Традиционный менеджмент пытается вобрать Agile-подходы и развиваться

2000 2010 2017

Подробности – мой доклад
«Agile – ответ на вызовы третьей промышленной революции»

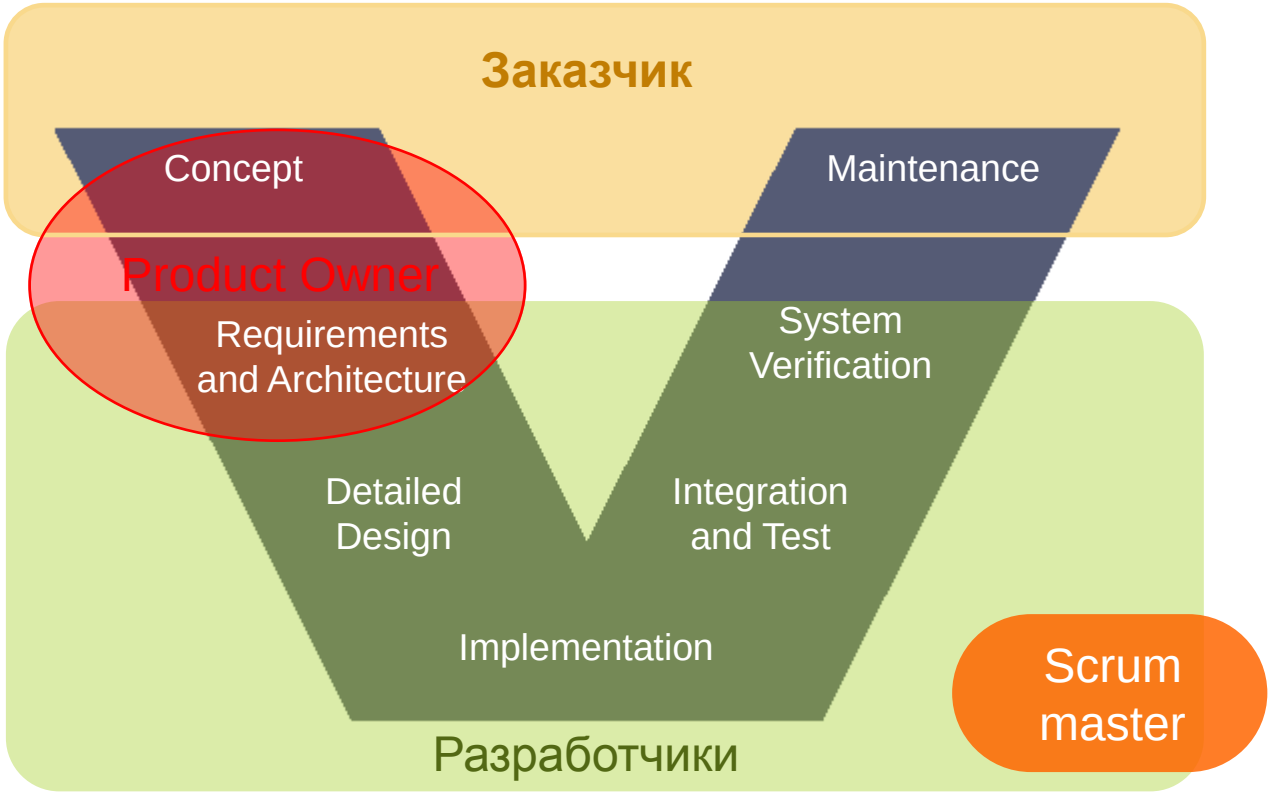
В команде – только разработчики

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



Плюсы и минусы

- ☺ Эффективные коммуникации
- ☺ Возможность быстрой обратной связи
- ☹ Большая нагрузка на Product Owner’а
- ☹ Слишком разнообразная работа членов команды

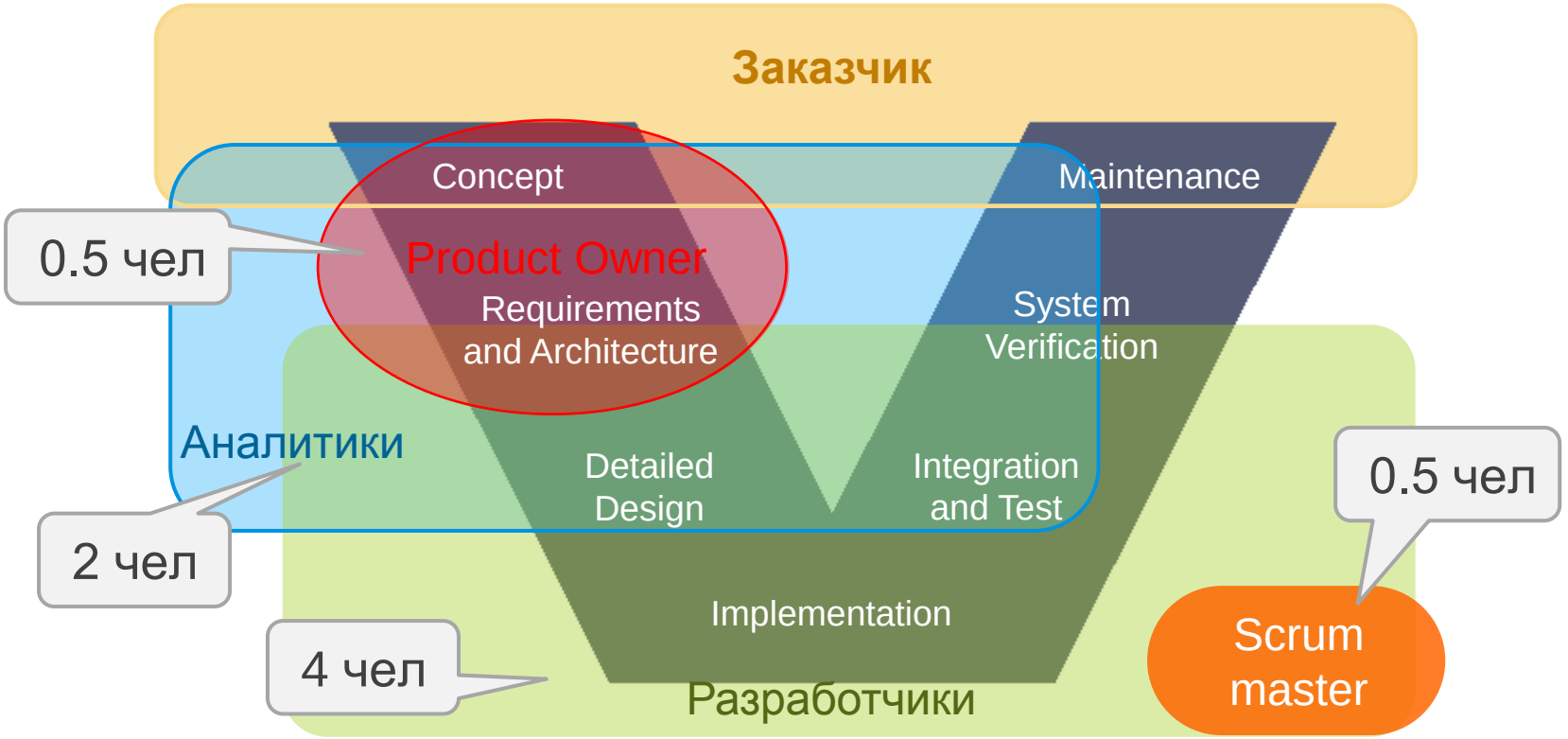


Аналитики устраняют разрыв

Вопрос: Насколько аналитики заняты в тестировании и сдаче системы?

Вариант 1: Сдают разработчики

Вариант 2: Аналитики тестируют и сдают **(модель внутреннего заказчика)**



Размер команды 7 ± 2

Модель внутреннего заказчика

- Аналитики получают требования заказчика и формулируют задачу разработчикам
- Они же принимают результат разработки и передают его заказчику

☺ Автономность команды разработки

☺ Эффективные коммуникации внутри и с заказчиком

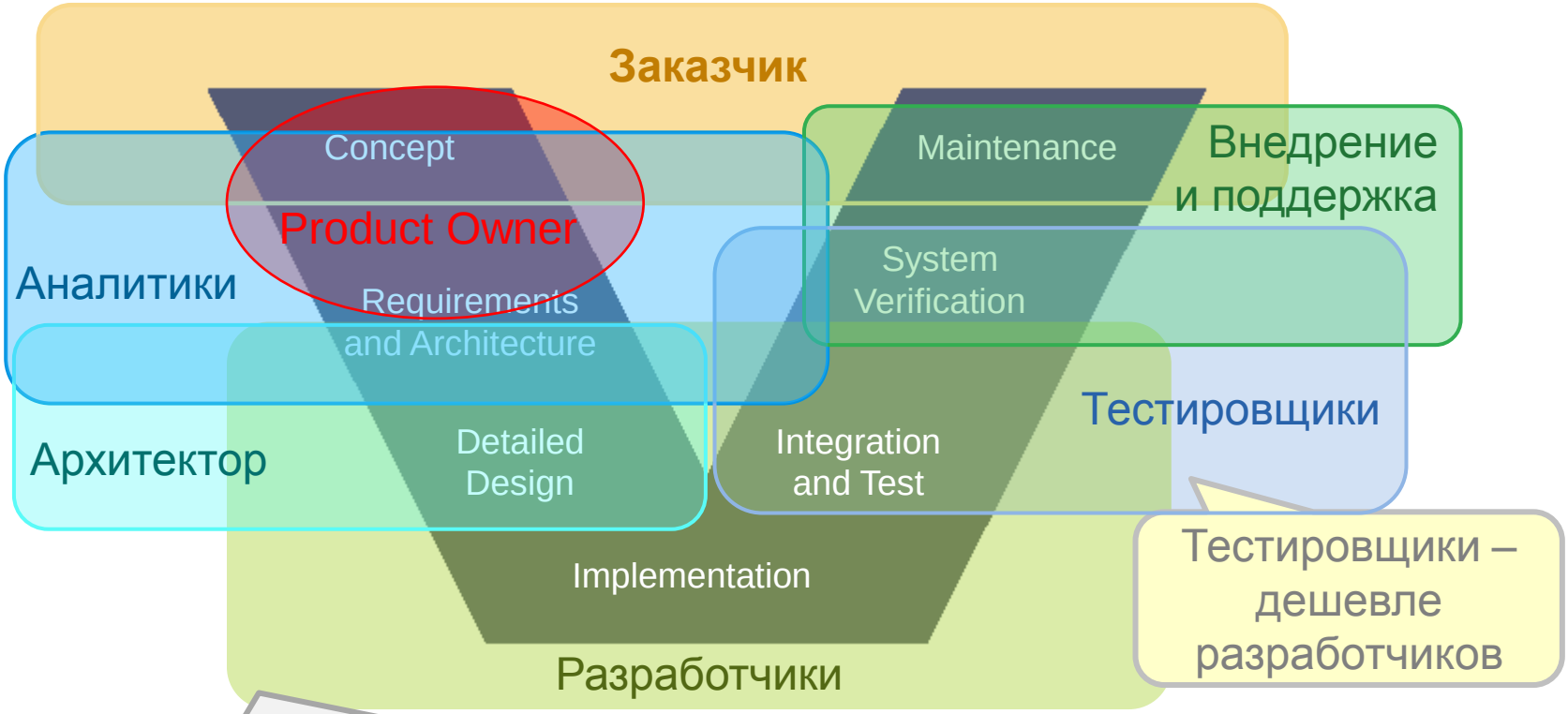
☺ Быстрая реакция на требования заказчика (скорость поставки часто важнее качества продукта)

☺ Прием результата разработки аналитиком повышает соответствие системы ожиданиям заказчика

☺ Две роли в команде – возможность дублирования

☺ Равномерная нагрузка на роли в ходе проекта

Дальнейшая специализация

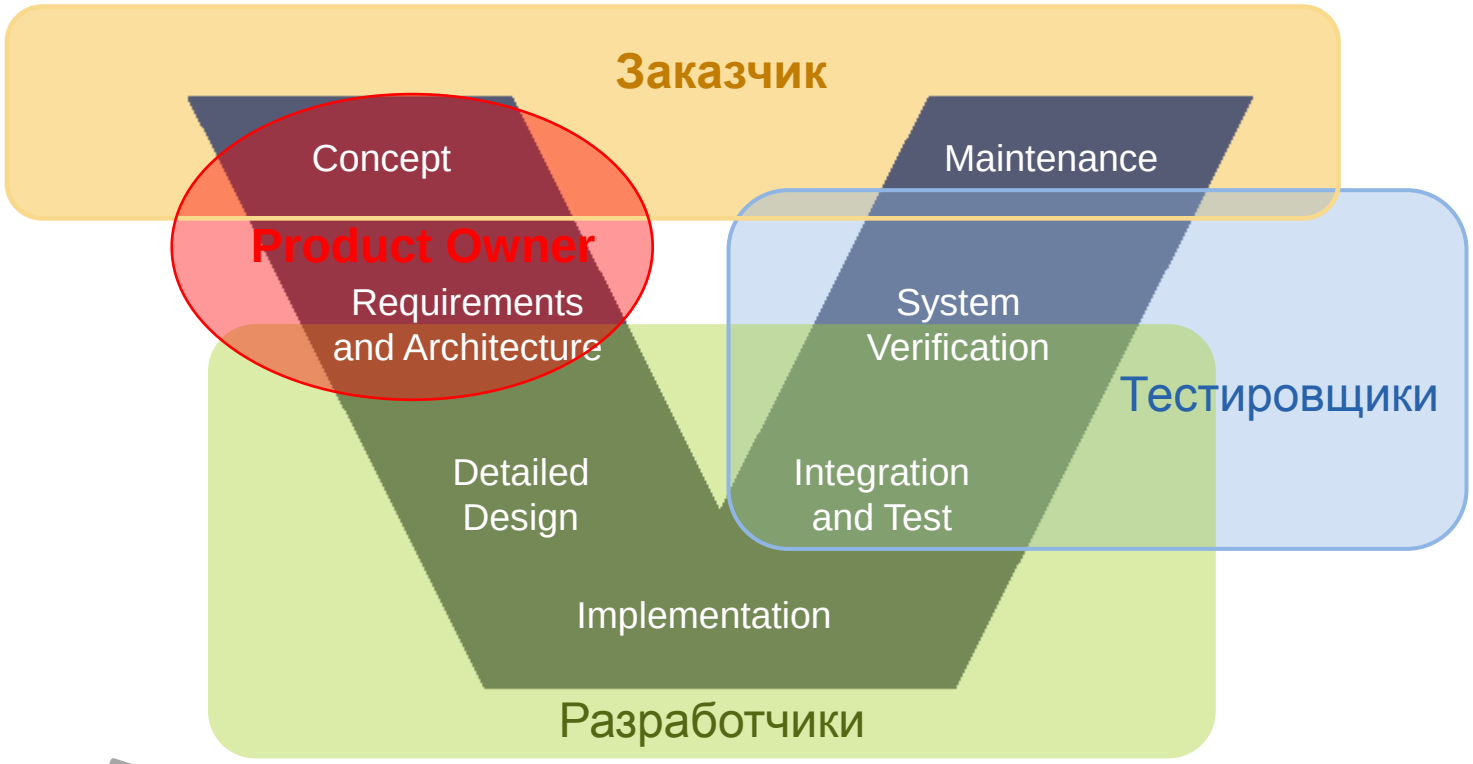


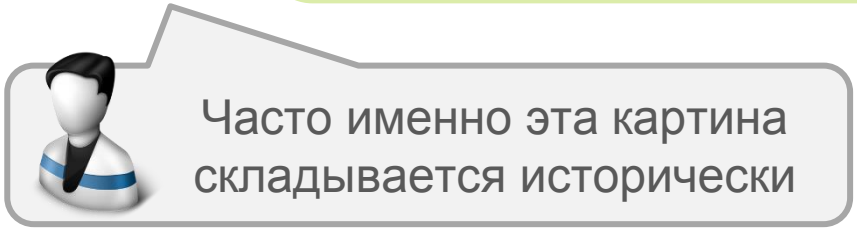
Мягкая кроссфункциональность: команда не должна быть заложником одного человека из-за специализации и эффективно обрабатывать **неоднородный** поток задач

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, характеризуемых 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

Простая команда без аналитиков

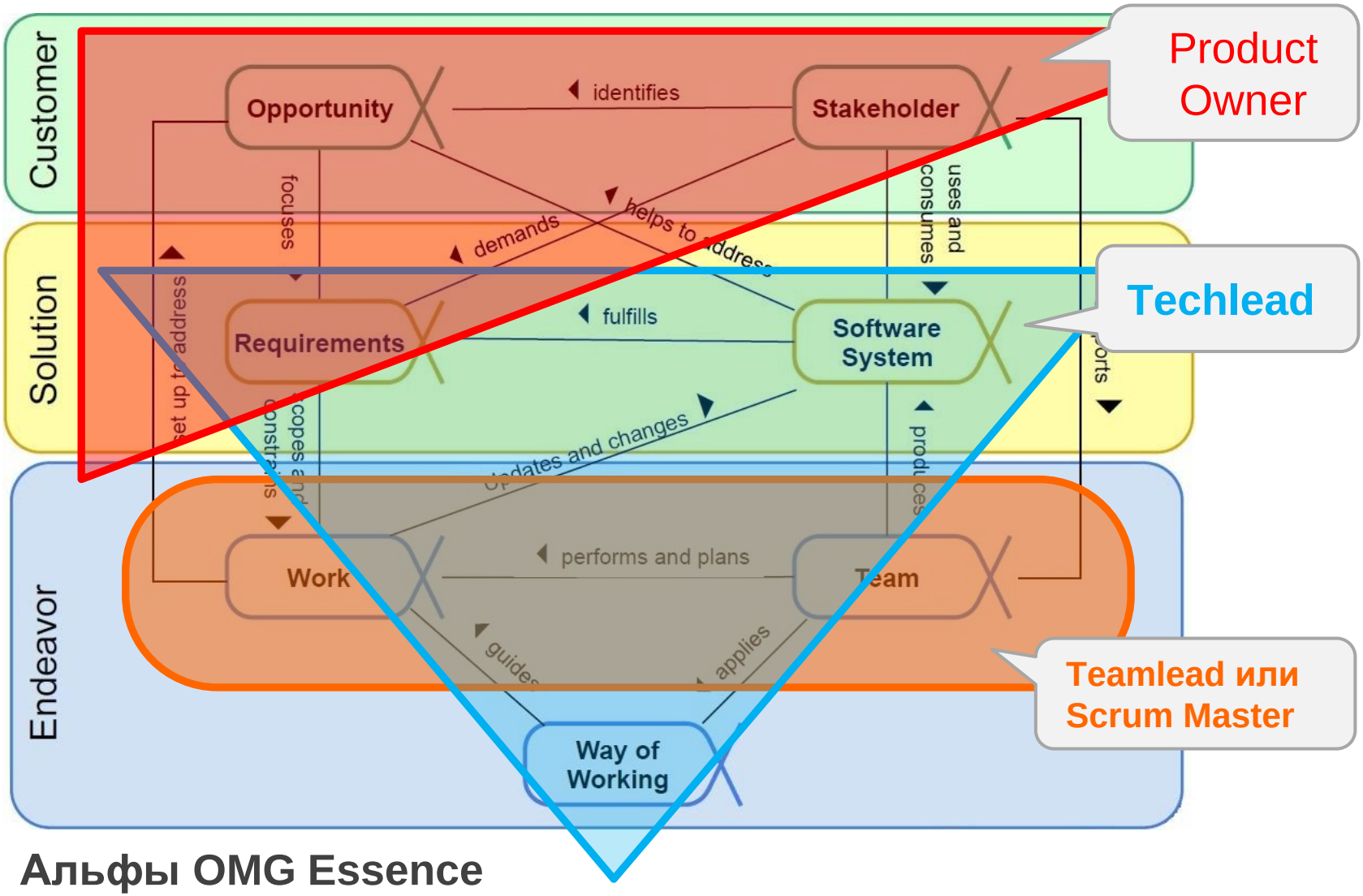




Ответственность за управление

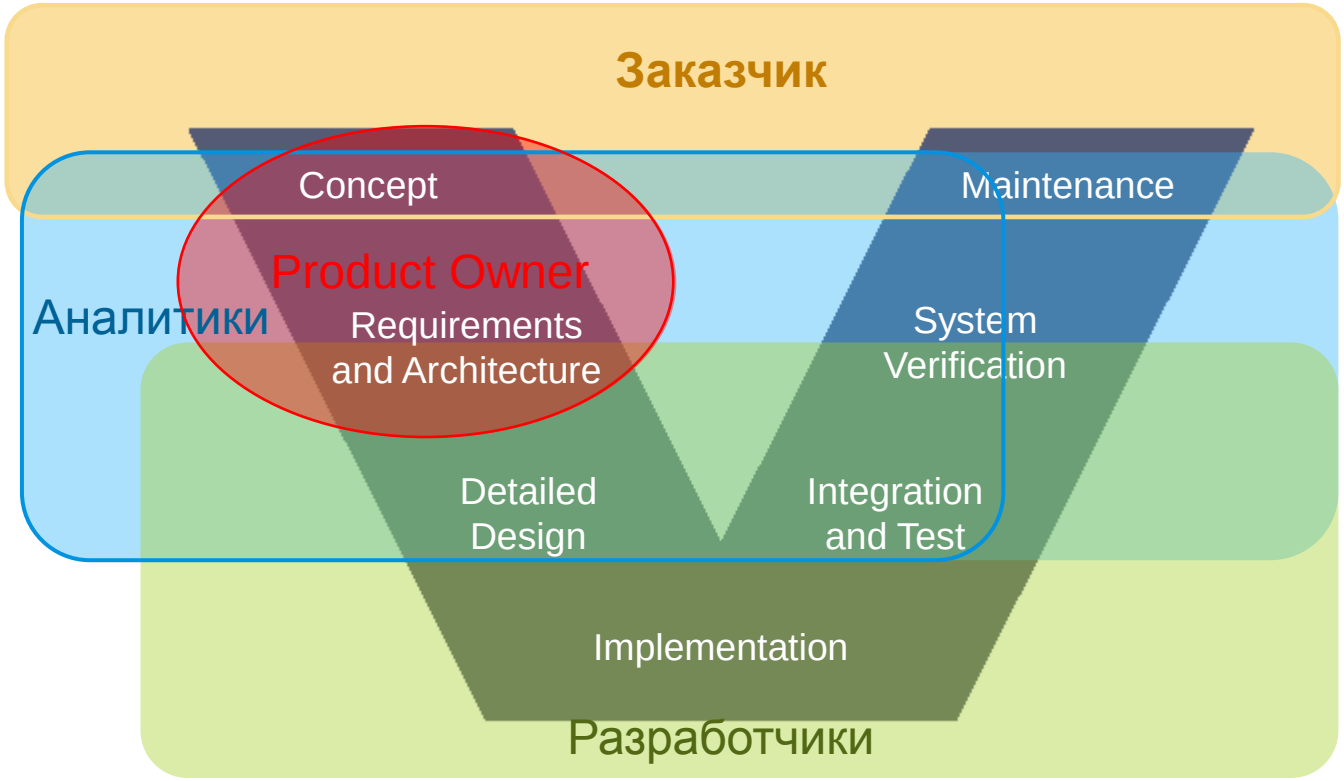
- Классический подход: за все отвечает менеджер
- Agile разделяет ответственность
  - Product Owner отвечает за продукт
  - Команда самоорганизуется для выполнения работ
  - Scrum master помогает эффективной самоорганизации
  - Менеджер тоже есть, но на следующем уровне управления
- Если квалификации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

Ответственность за управление



Альфы OMG Essence

Размер команды 7 ± 2

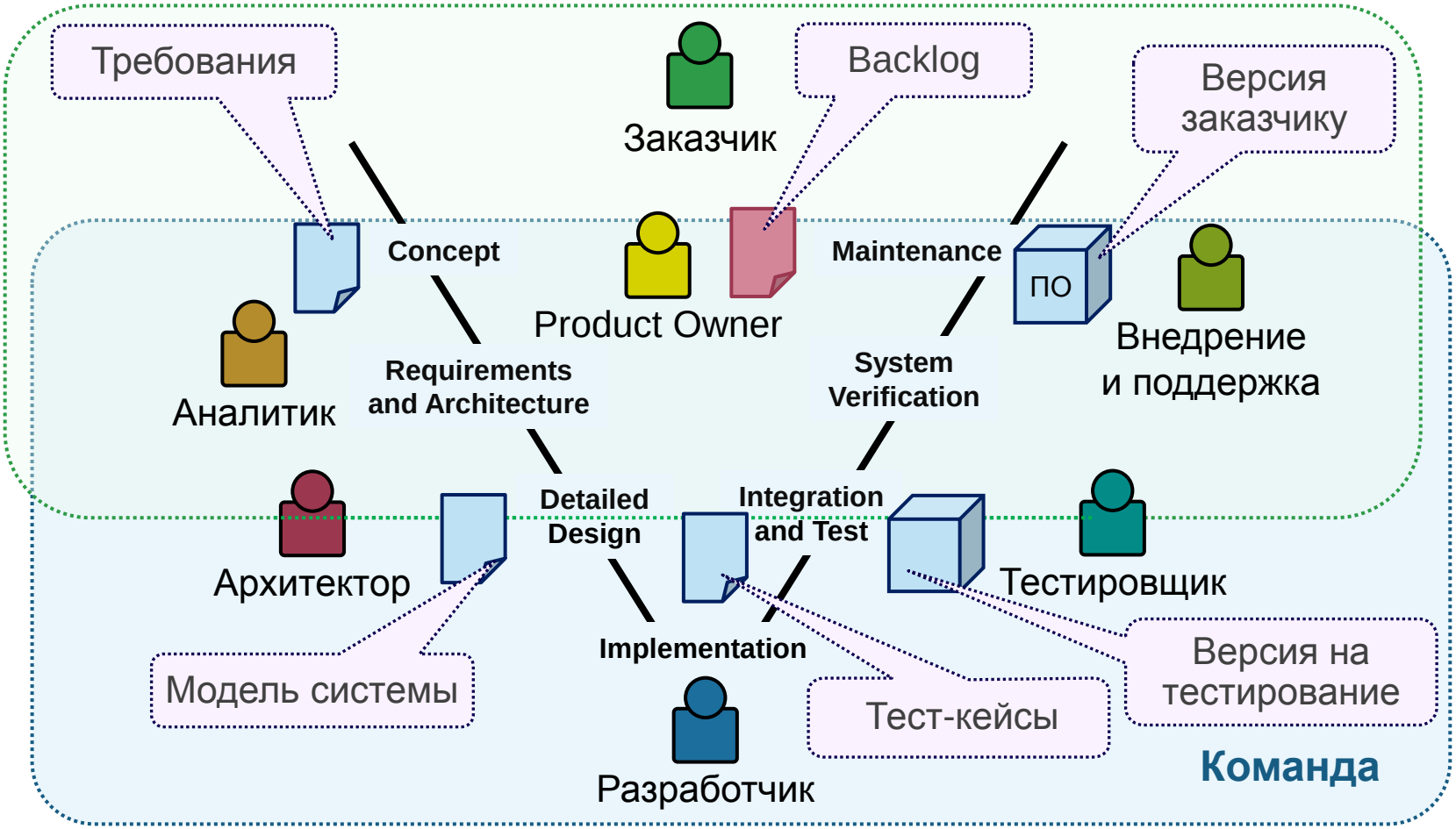


Артефакты – для коммуникации и сохранения знаний

Принципы работы с артефактами

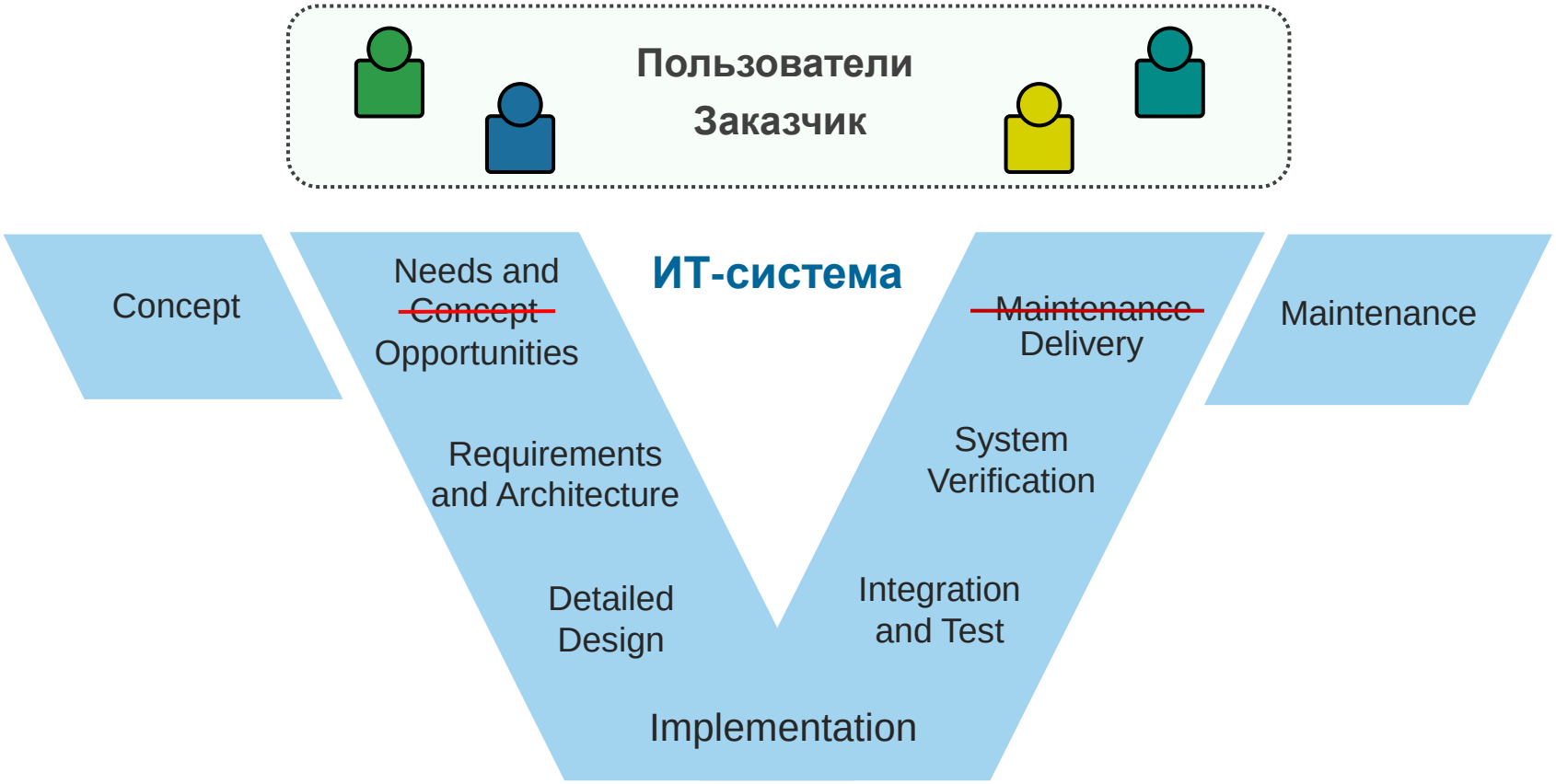
- **Артефакт – средство коммуникации**
  - С заказчиком и разработчиками в ходе проекта
  - С будущими пользователями системы
  - С теми, кто будет развивать и эксплуатировать – даже если это ты сам, но через полгода
- Артефакт должен быть понятен всем сторонам коммуникации
  - Это ограничивает сложность нотаций
- Упрощенные схемы должны сохранять ключевые моменты

Ответственность за артефакты

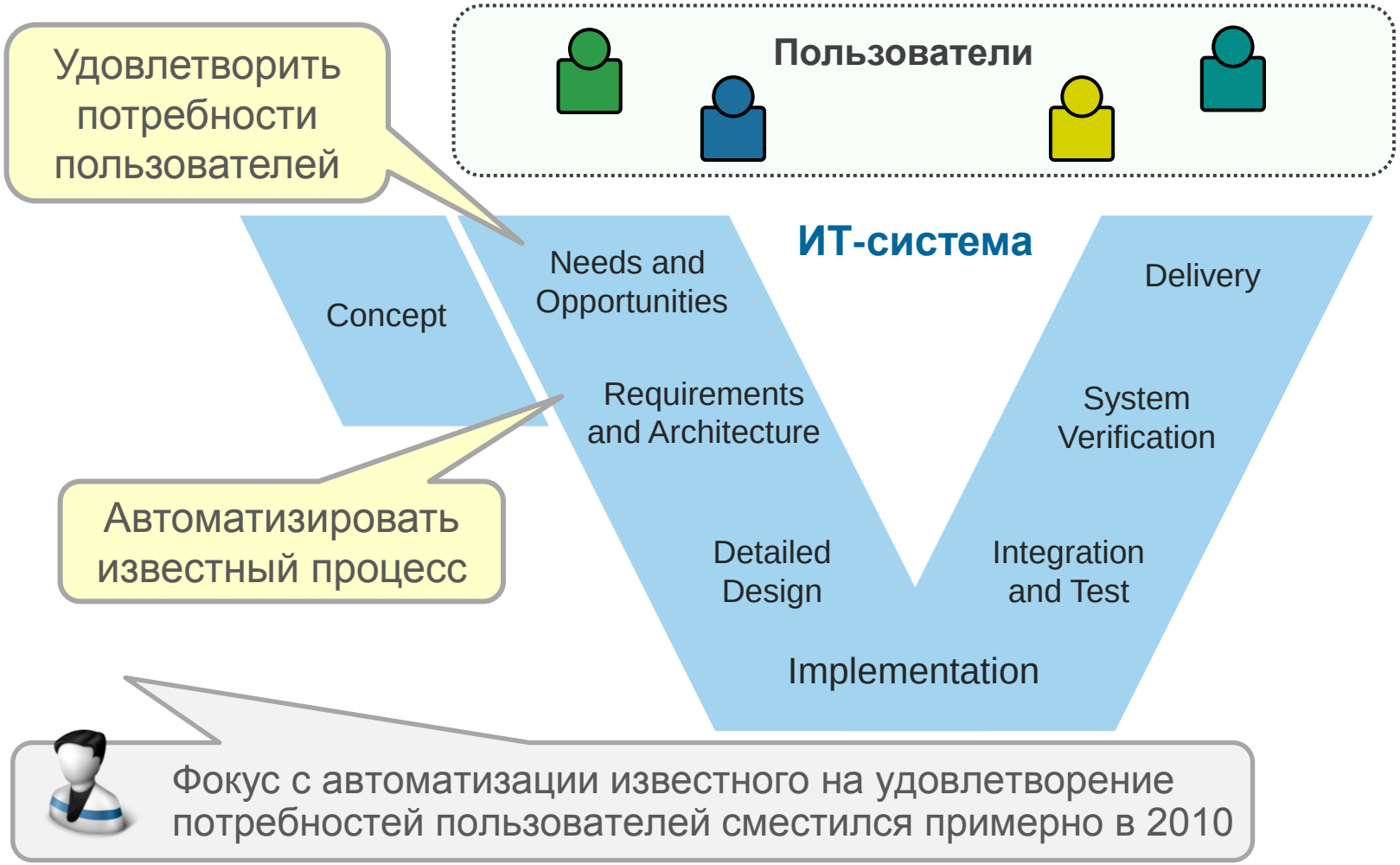


Границы поляны ответственности

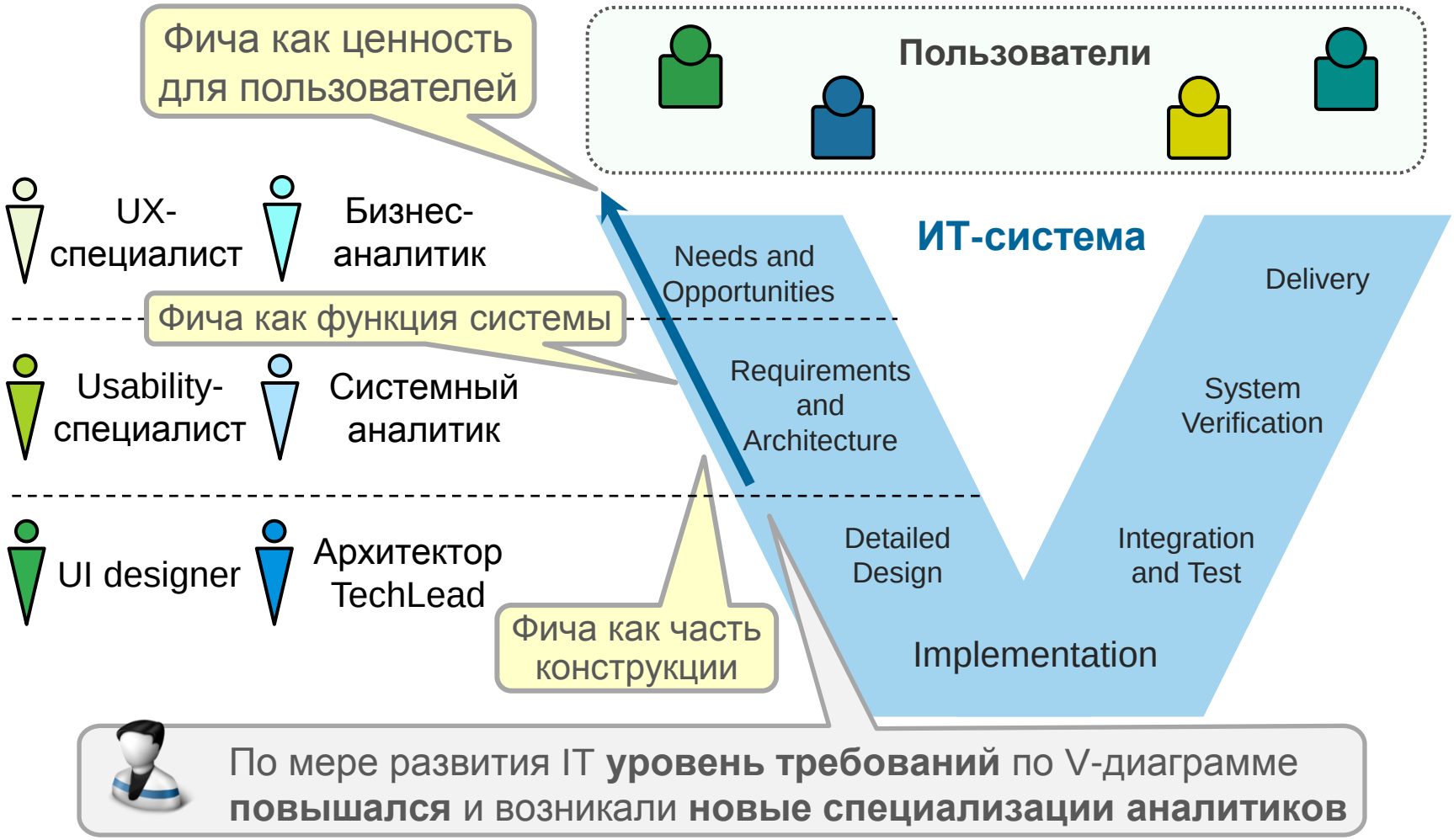
Расширенная V-модель



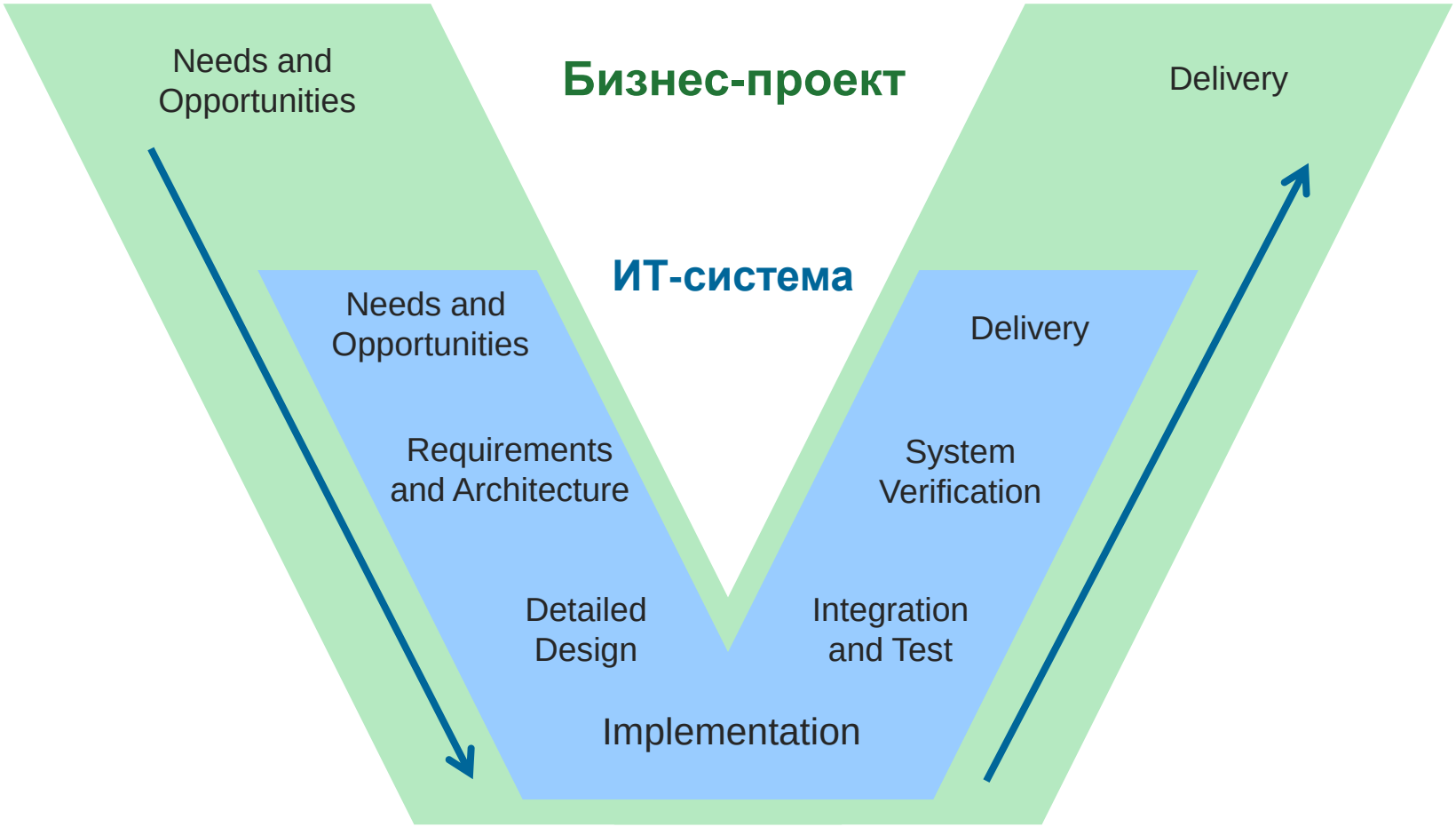
Что является целью проекта?



Различная внешняя граница проекта

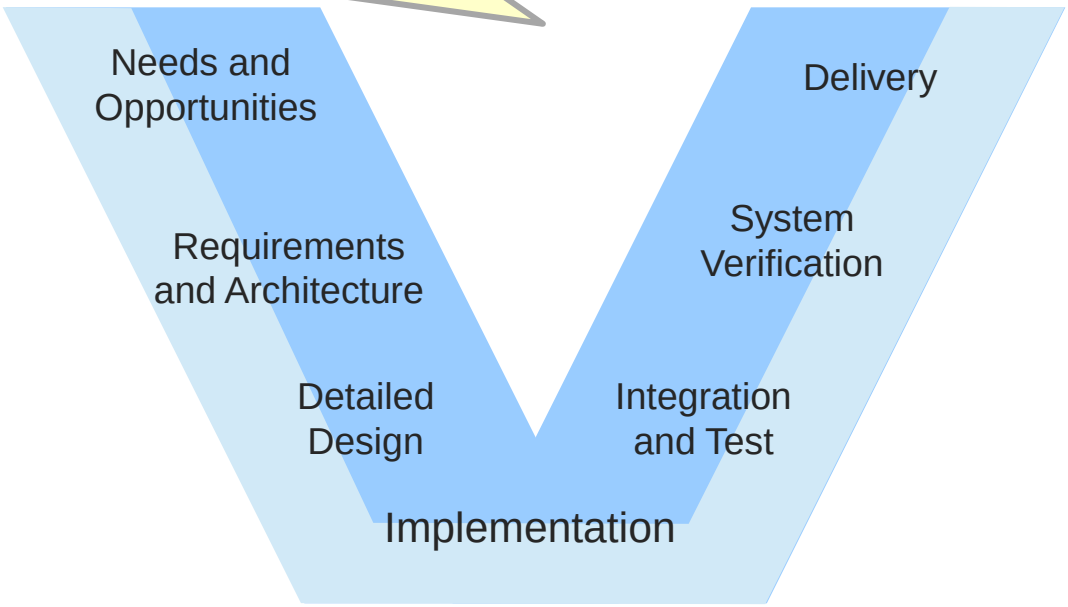


IT-проект внутри бизнес-проекта?



Или IT и бизнес делают **совместный проект**?

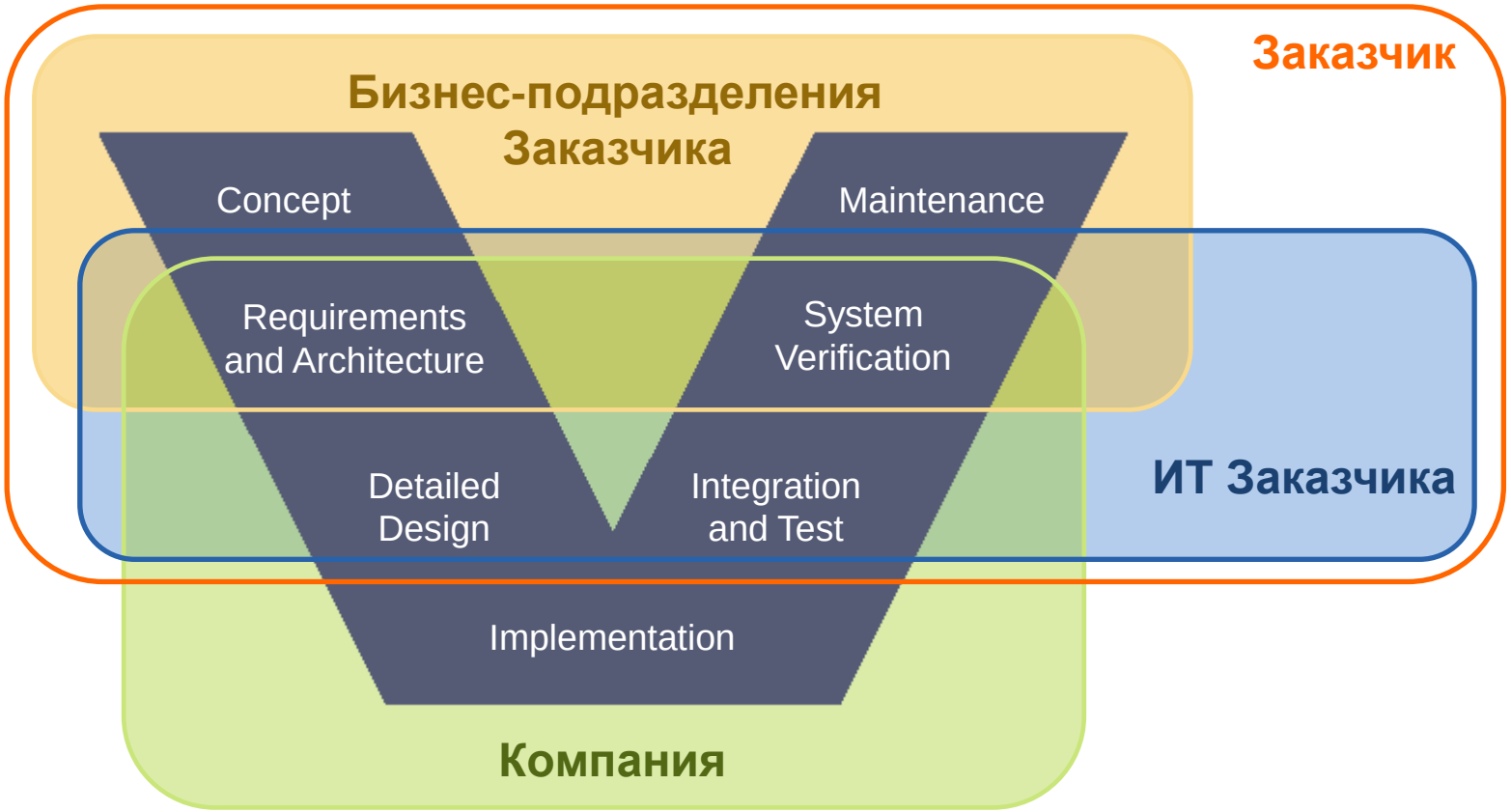
В современных проектах граница между IT-частью проекта и изменением бизнеса подвижна и меняется в ходе проекта, а **диджитализация** означает, что **проект будет единым, а IT-часть – главной!**



Сколько у проекта команд?

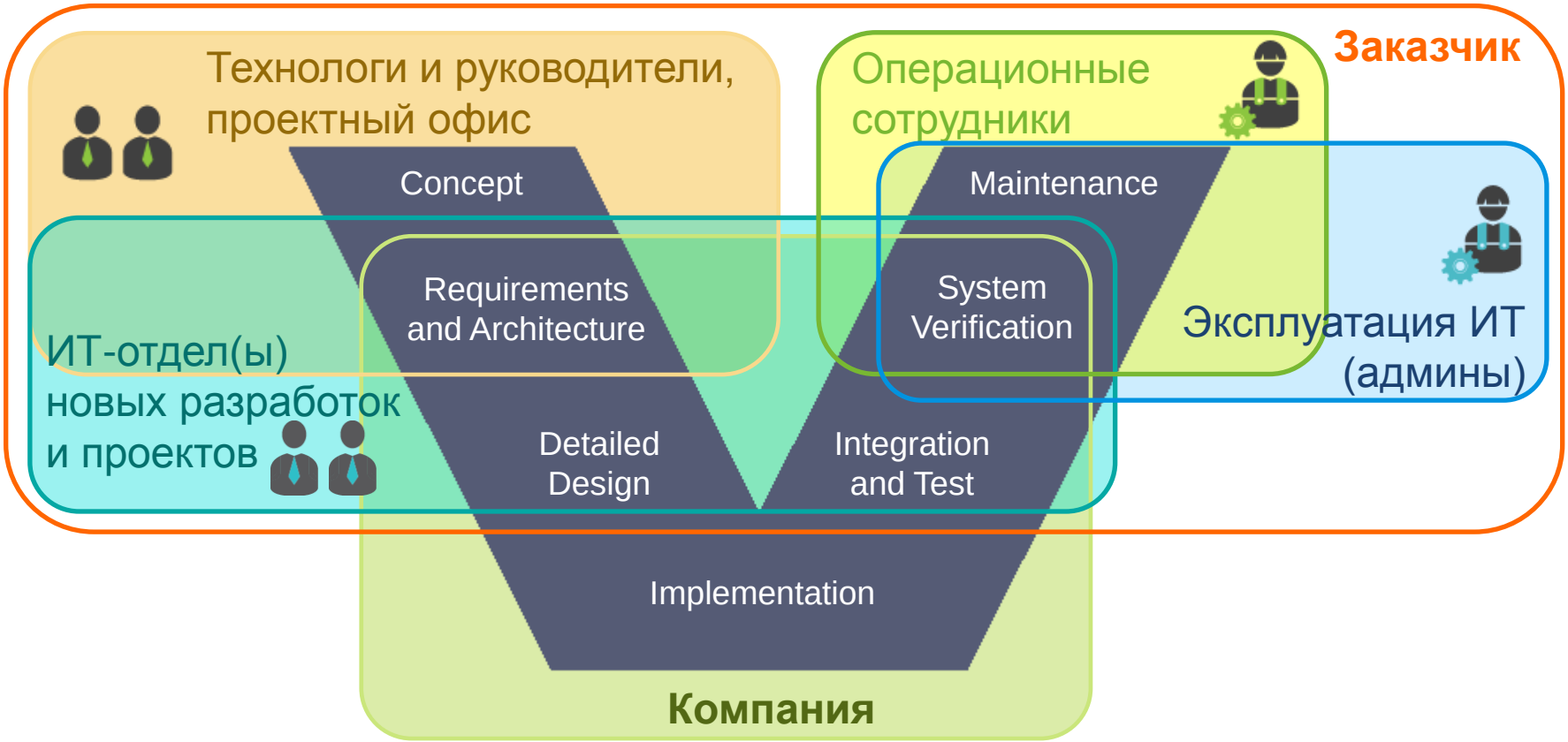
У заказчика есть свой ИТ…

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

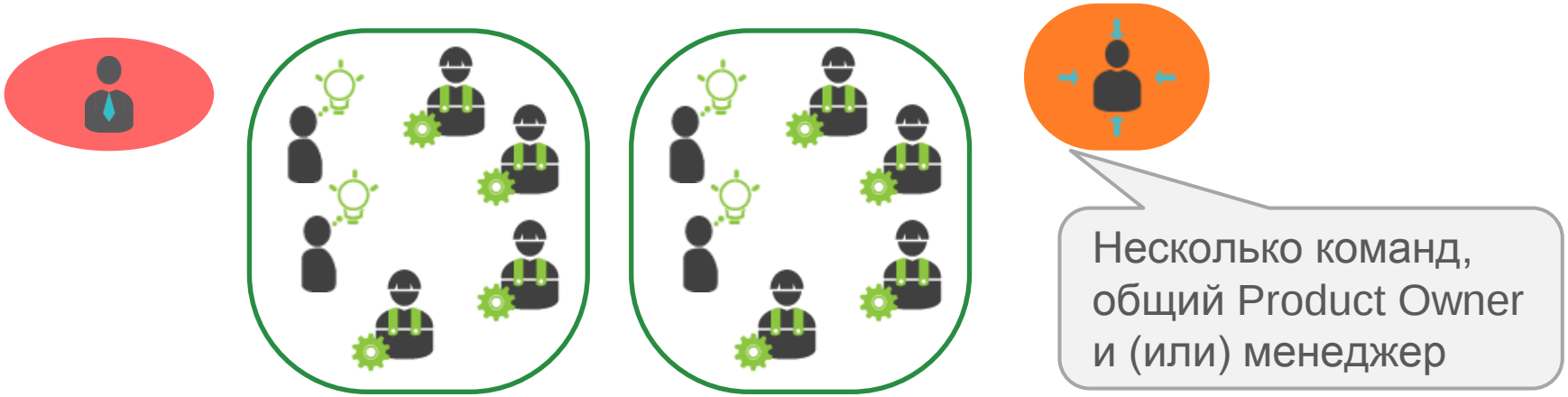


Операционная работа и развитие

Постановку задачи на разработку и эксплуатацию созданного приложения осуществляют разные группы стейкхолдеров



А если проект большой?

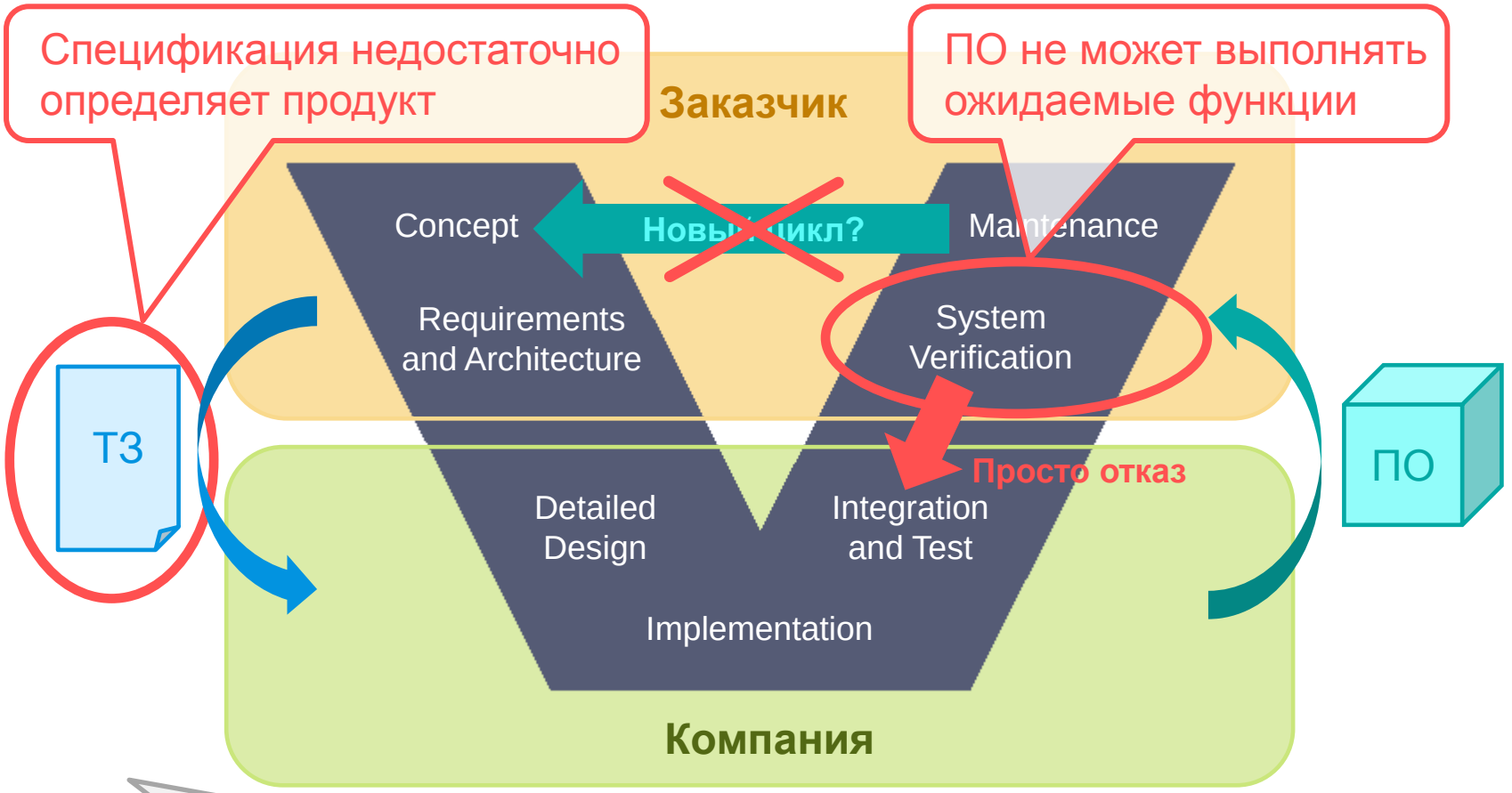


Аутсорсинг отдает управление

- Компания не организует работу специалистов
- Компания не подбирает специалистов
- Компания не поддерживает компетенции

- Необходимо фиксировать границу ответственности
- И понимать риски…

Аутсорсинг кодирования



Помните уроки Agie: взаимодействие **только** через артефакты работает плохо, необходимы **коммуникации**, **инкрементальная поставка и обратная связь**

Аутсорсинг тестирования и поставки

- Тестирование и организация конвейера поставки – специализированные функции, отдельная компетенция
- Аутсорсинг имеет смысл, если разработку ведут несколько компаний – обеспечивается интеграция
- Или в случае высоких требований к технологиям, которые не позволяют держать собственного специалиста
- Необходимо хорошо определять границы и передачу ответственности

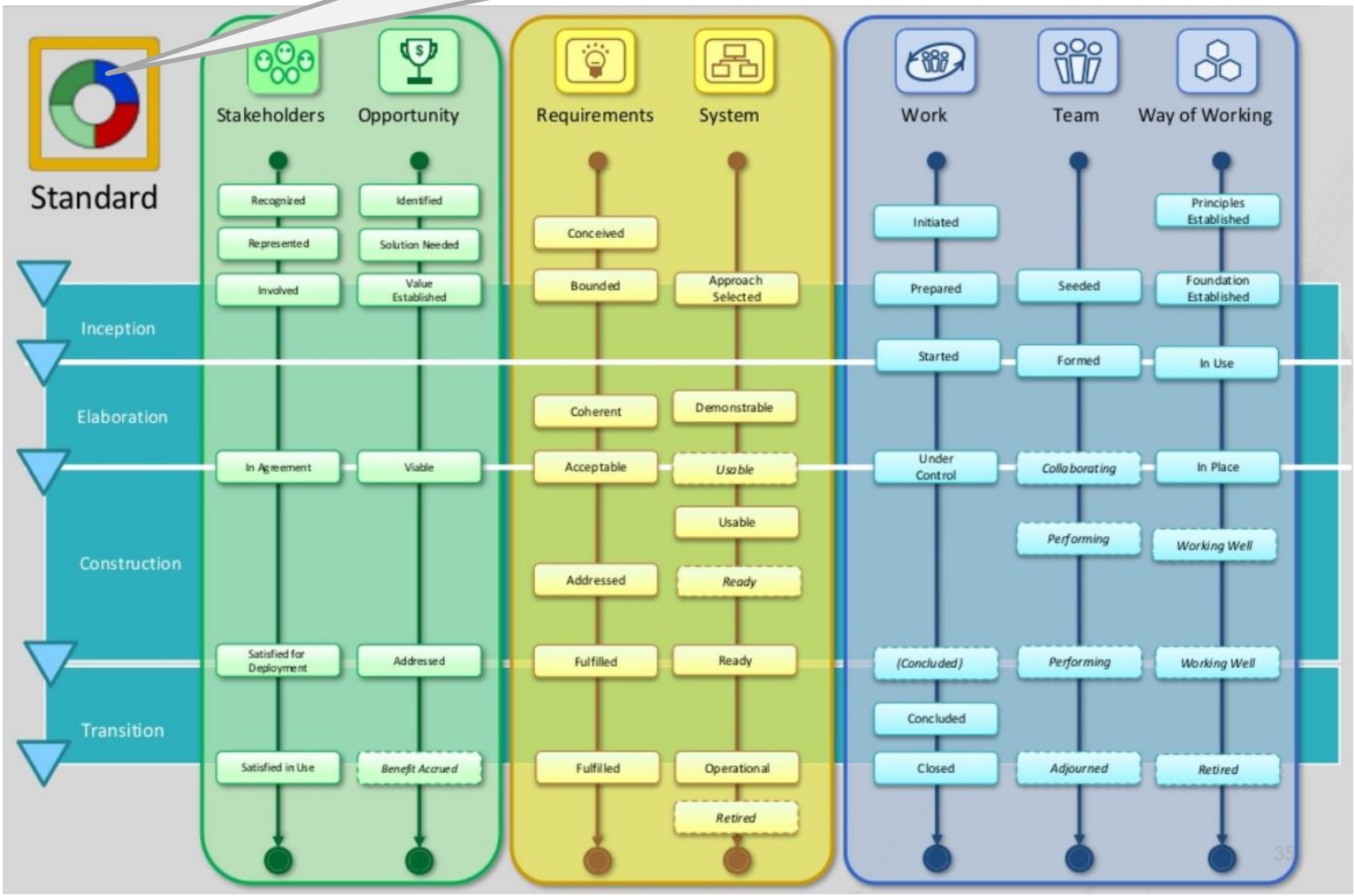
Детальное разделение ответственности

Используем Lifecycle OMG Essence

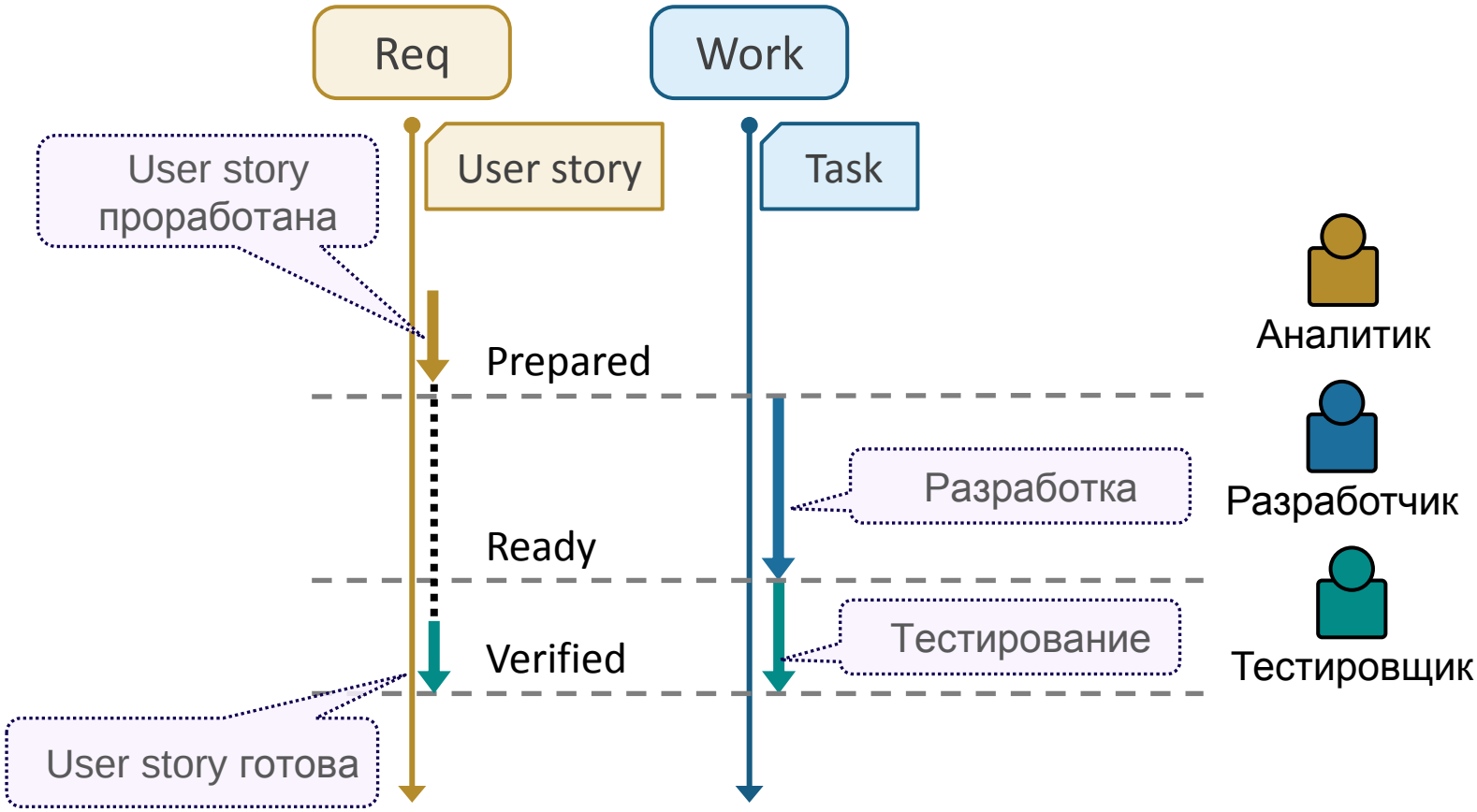
- Альфы задают объекты, состояния которых отражают движение проекта
- Ответственность делится по этим объектам
- Check lists состояний определяют, в чем ответственность
- Lifecycle-диаграмма иерархична: можно представить весь проект, его релиз, спринт или разработку фичи

Артефакт – экземпляр объекта

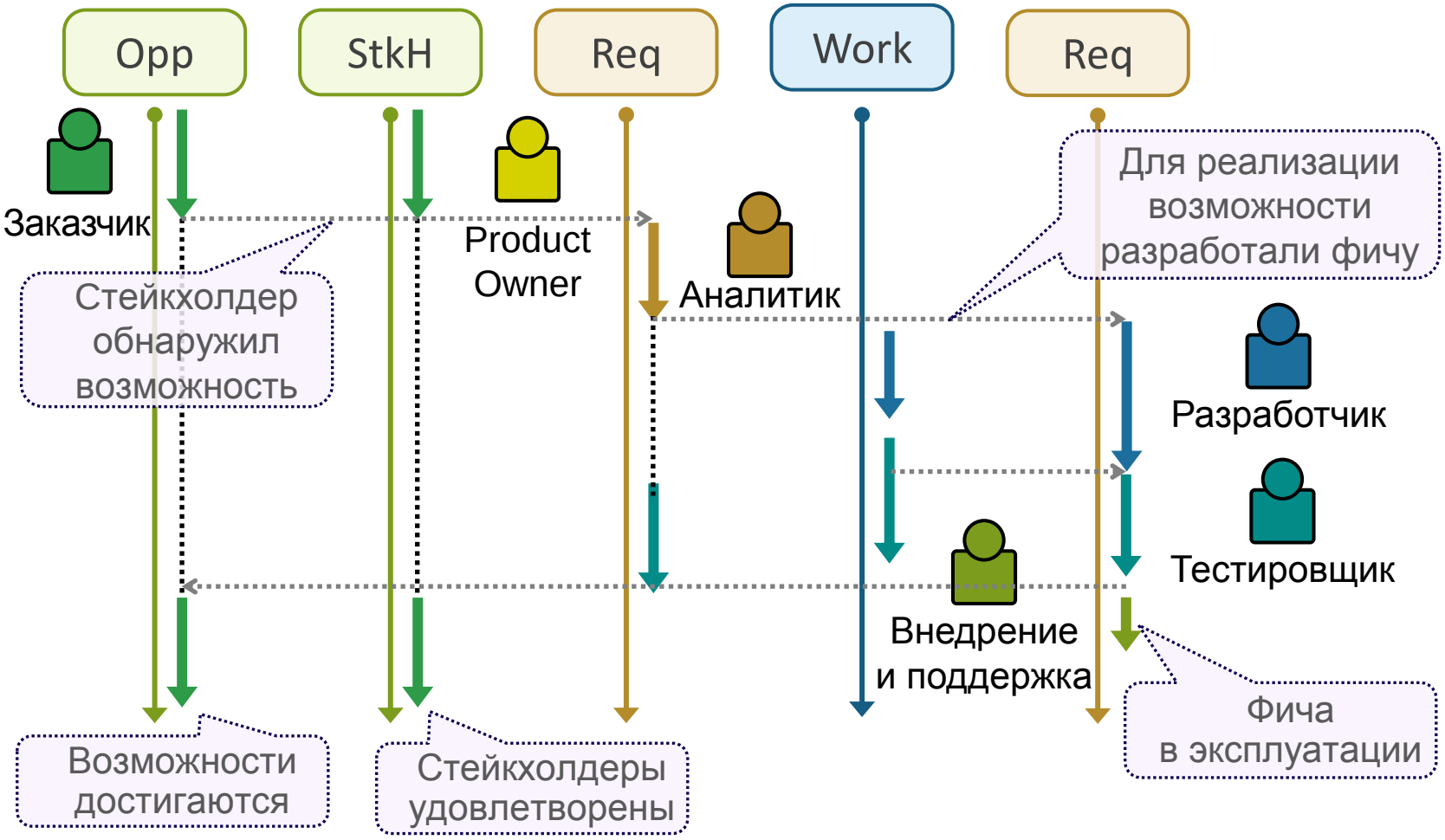
Пример: жизненный цикл «стандартного» проекта



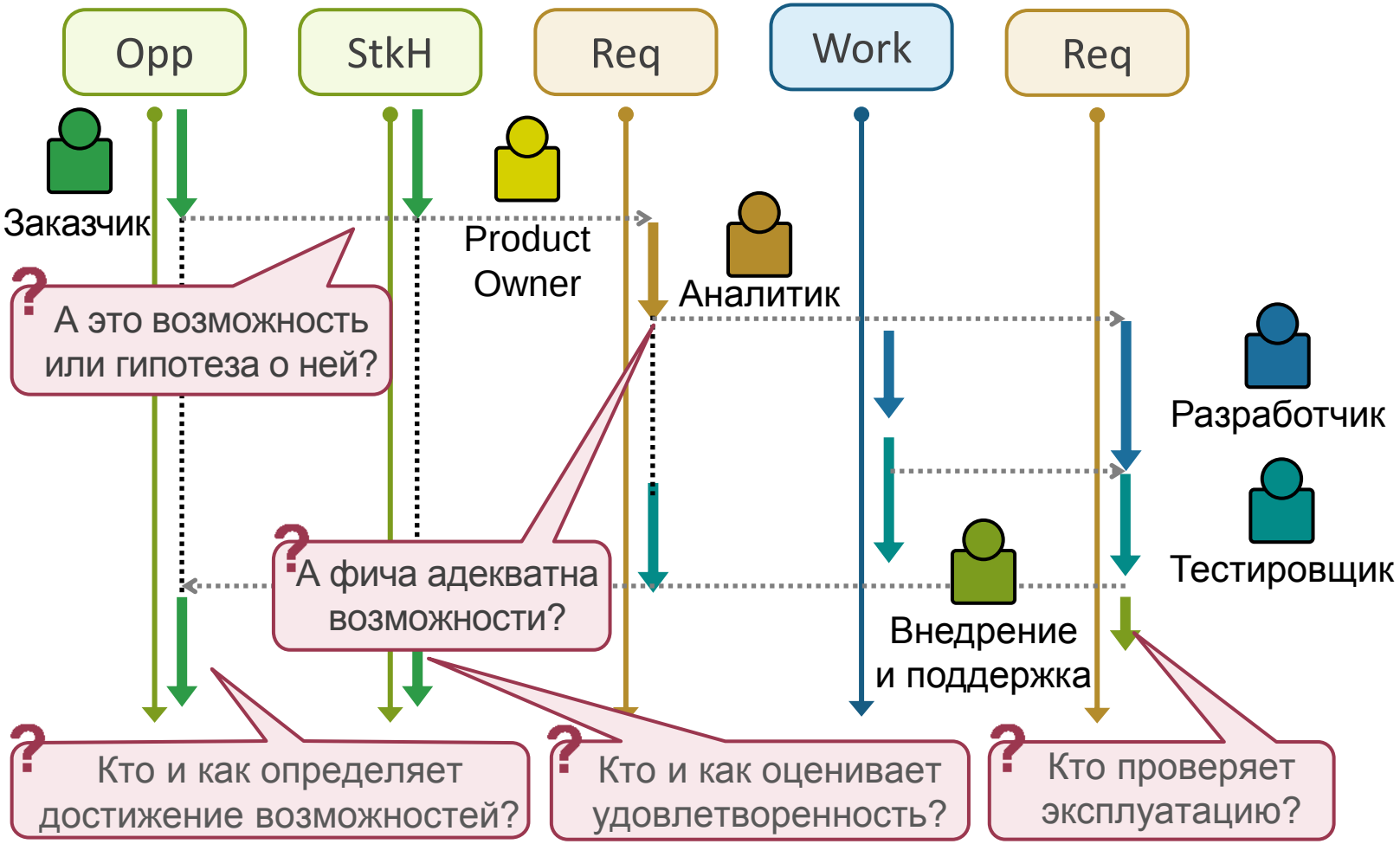
Простейший вариант – ответственность за задачу



Удовлетворенность стейкхолдеров и обеспечение возможностей бизнеса



Кто проверяет цели и их достижение?



Заключение

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

Вопросы? Обращайтесь!

Максим Цепков http://mtsepkov.org