CUSTIS®

Разделение ответственности в заказной разработке

Максим Цепков

Главный архитектор
дирекции развития решений

18 апреля 2015 года

Minsk Analyst Days 2015

CUSTIS®

О чем будет доклад

- Разделение ответственности в проекте – популярная тема холиваров
  - Многие уверены, что знают «правильный способ»
  - Часто выдают претензии смежникам, что те не делают положенное или, наоборот, лезут на чужую поляну
- Однако единственной схемы не существует, это пережиток эпохи «правильного процесса»
- Разделение ответственности нужно конструировать в проекте, с учетом его особенностей и интересов участников

CUSTIS®

Содержание доклада

1. Схема для разделения ответственности
2. Простой кейс: разработка по спецификациям
3. Заказчик и компания-разработчик: разделение ответственности и взаимодействие
4. Ответственность в компании-разработчике

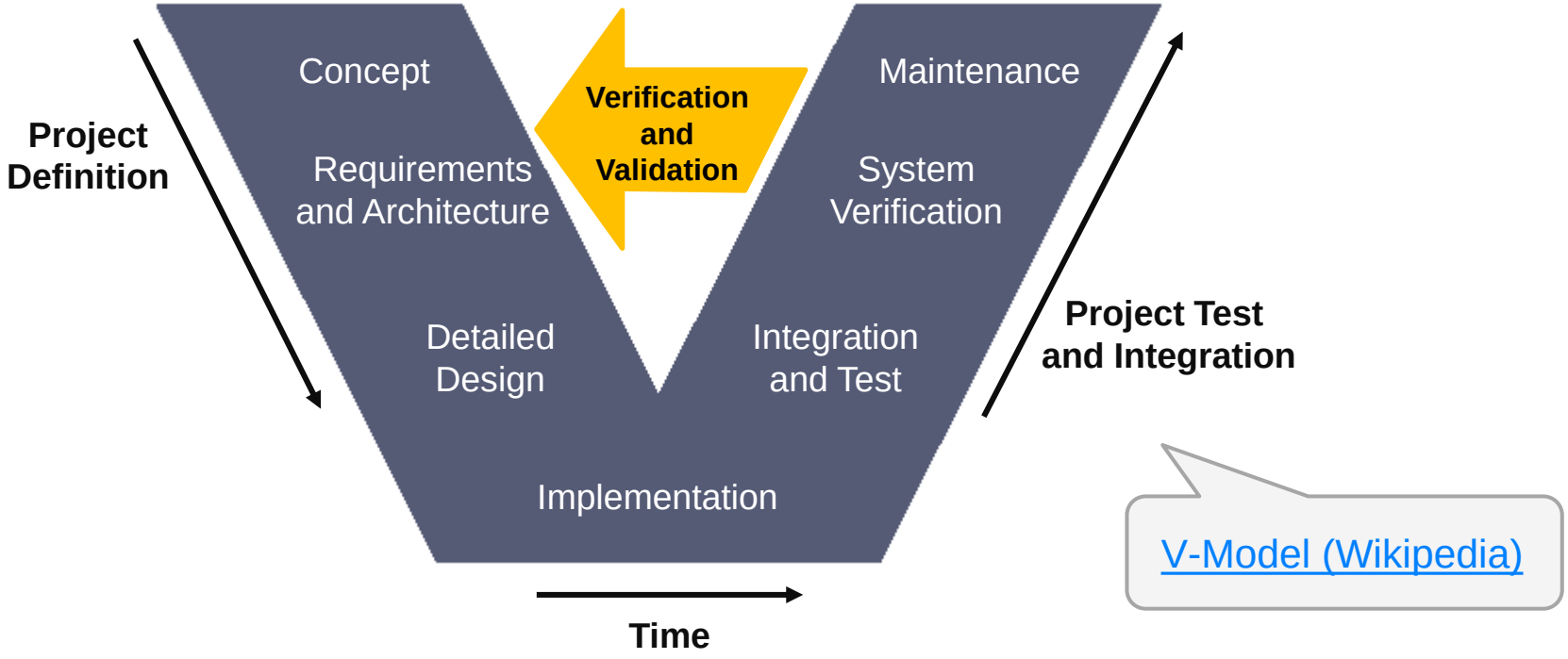
CUSTIS®

Схема для разделения ответственности

CUSTIS®

Визуальное представление

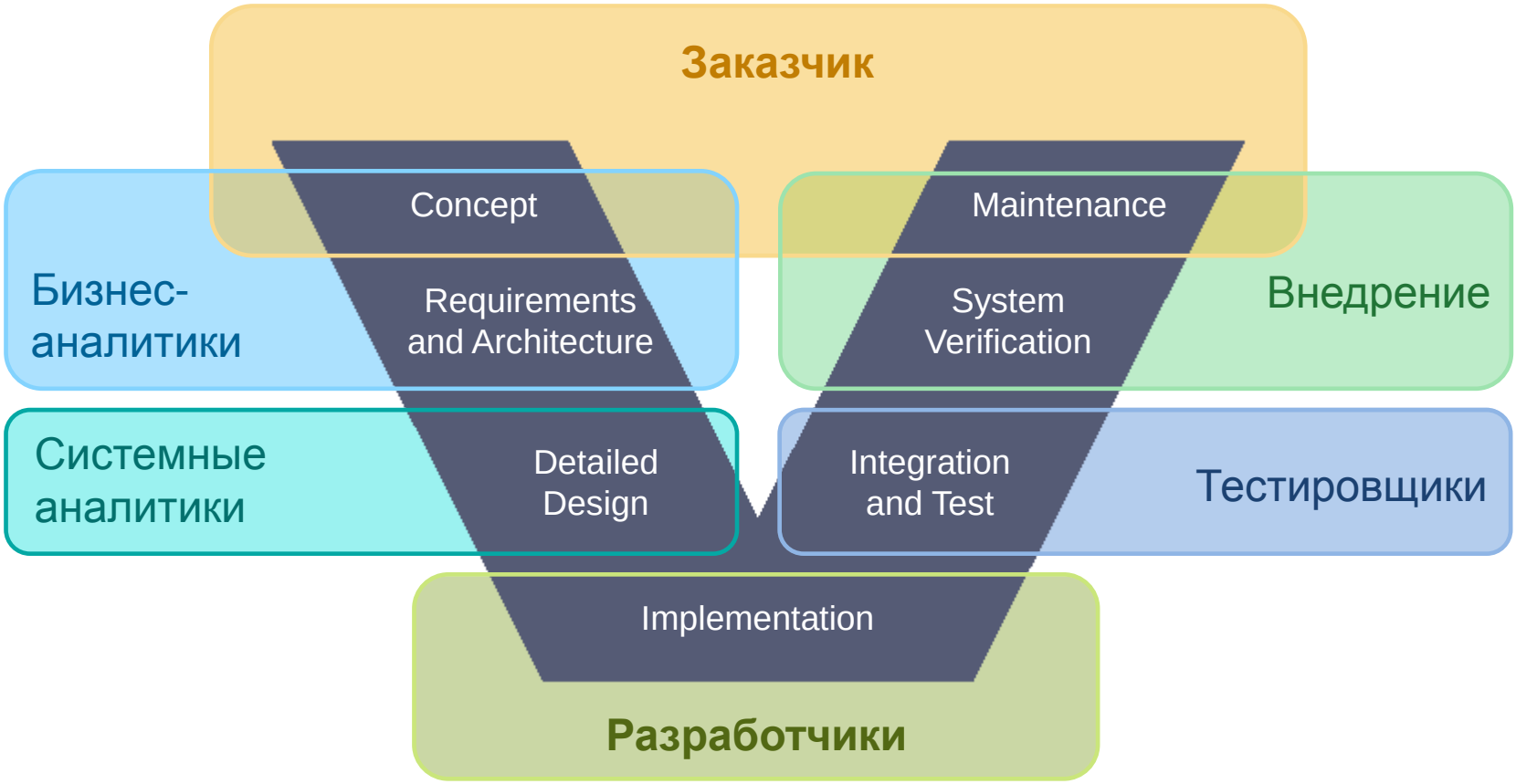
- Возьмем схему процесса – и разметим
- Используем стандартную схему – V-модель



V-Model (Wikipedia)

CUSTIS®

Водопадная модель на схеме



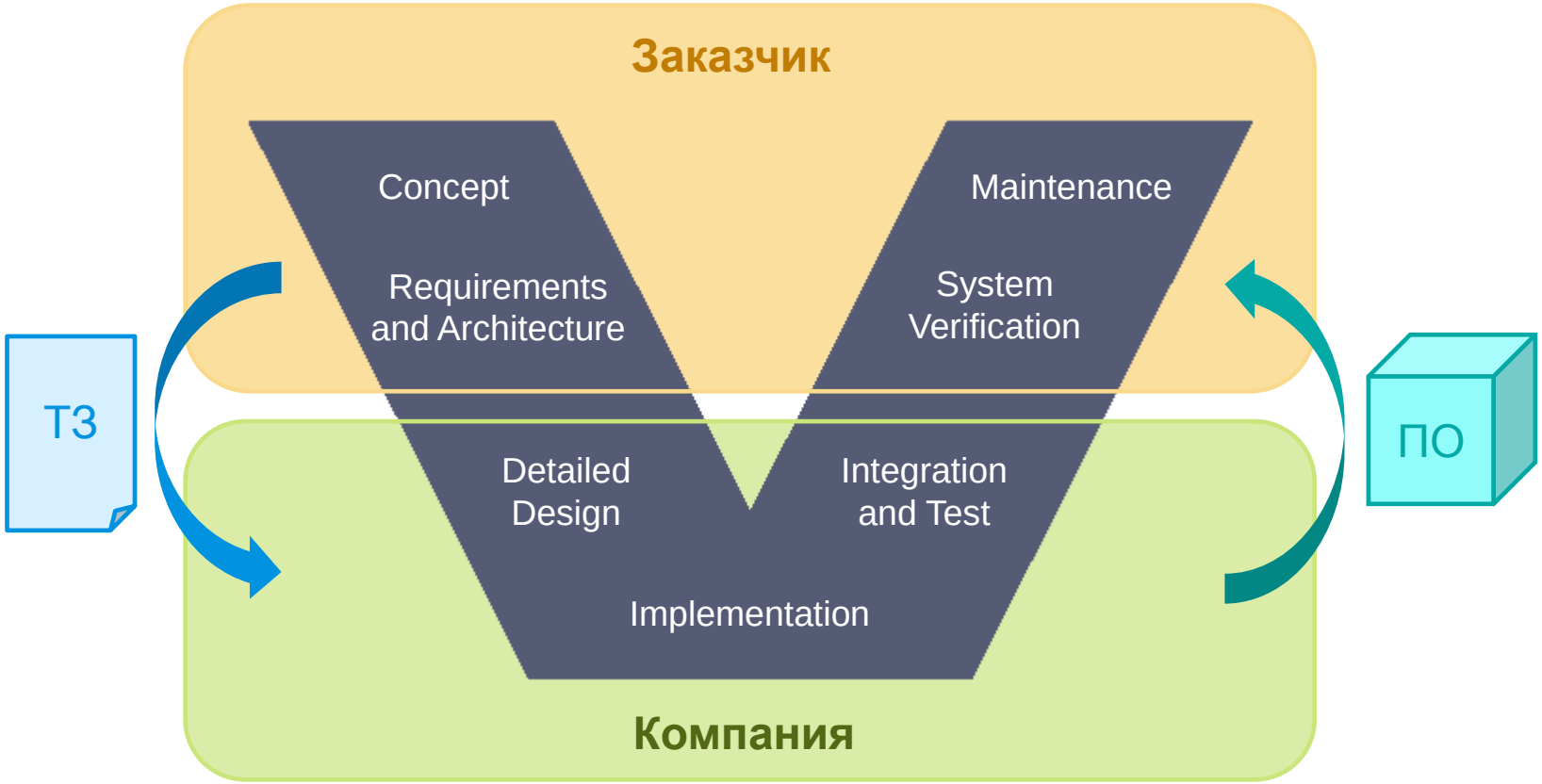
CUSTIS®

Простой кейс: разработка по спецификациям

CUSTIS®

Аутсорсинг кодирования

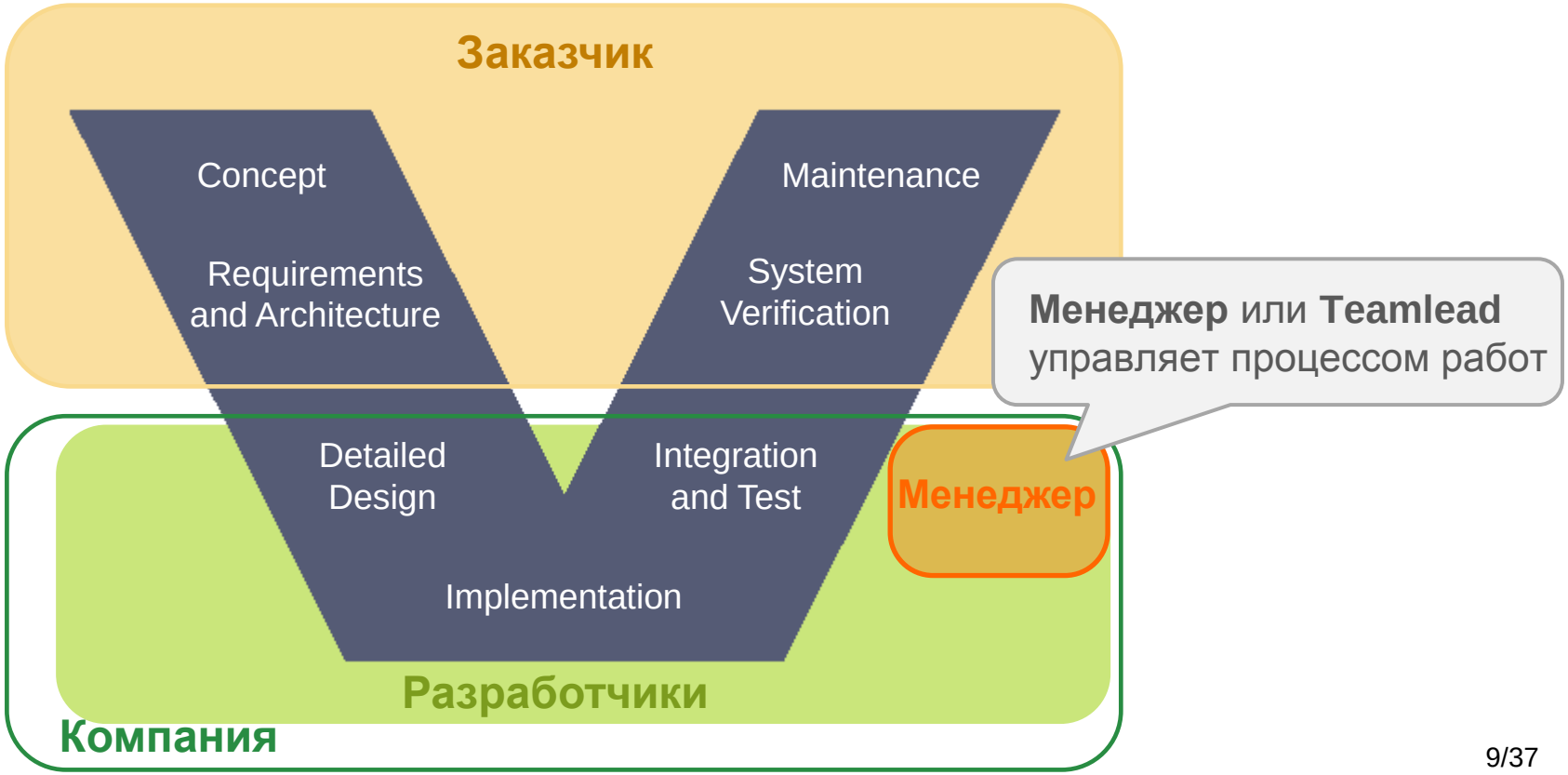
Уместен при высокой стандартизации продукта,
позволяет создать задание и определить результат



CUSTIS®

В компании – только разработчики

Работает, если проекты небольшие или имеют типовую декомпозицию, а также в случаях, когда применяются только типовые решения. Организация – **одна команда** или **мини-группы**

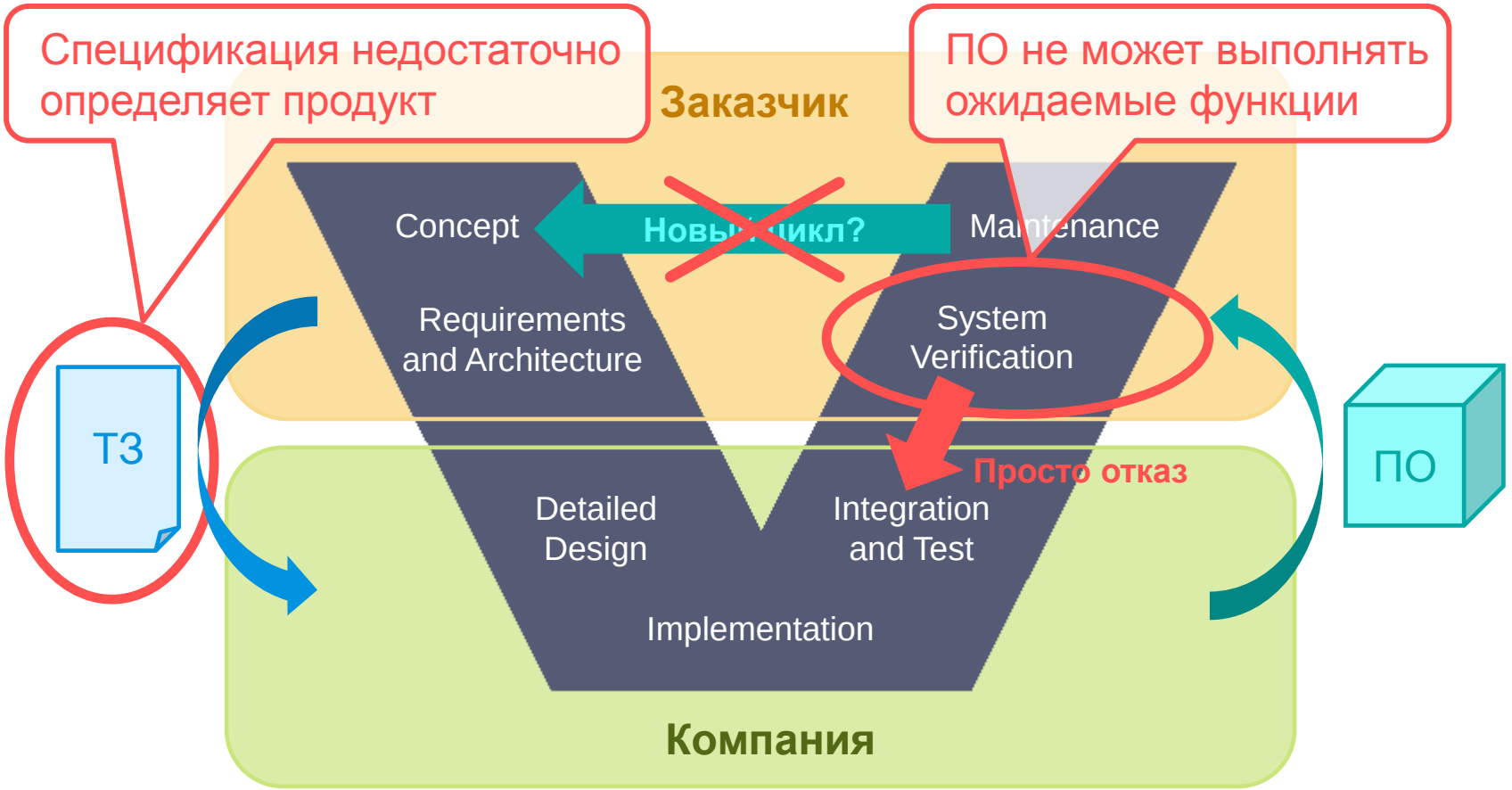


CUSTIS®

Специализация в команде

- По разным технологиям
  - Java и СУБД
  - Дизайнер и разработчик в web
- Выделение «дешевых» специализаций, например тестировщиков
- Работа может быть организована последовательно или параллельно
- При последовательной работе возможно выделение отделов
  - Например, дизайн – разработка – тестирование

Разрыв между заказчиком и компанией

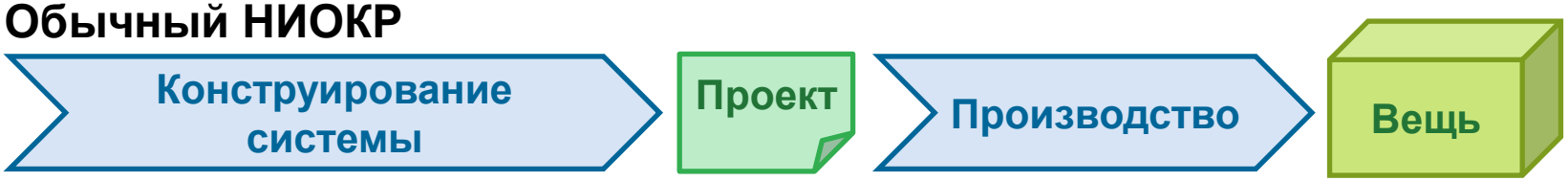


CUSTIS®

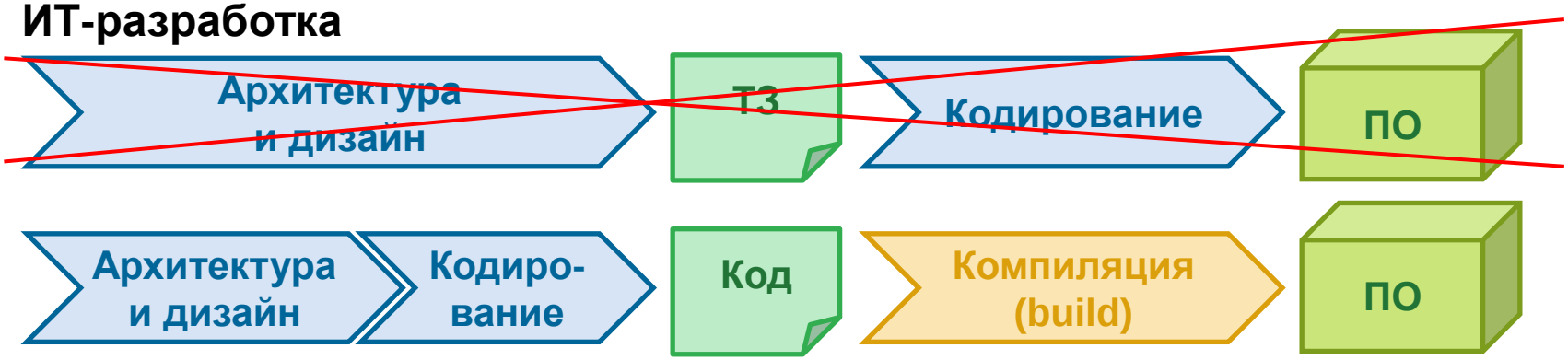
Почему плохо работает?

Мешает природа ИТ-разработки

Обычный НИОКР



ИТ-разработка



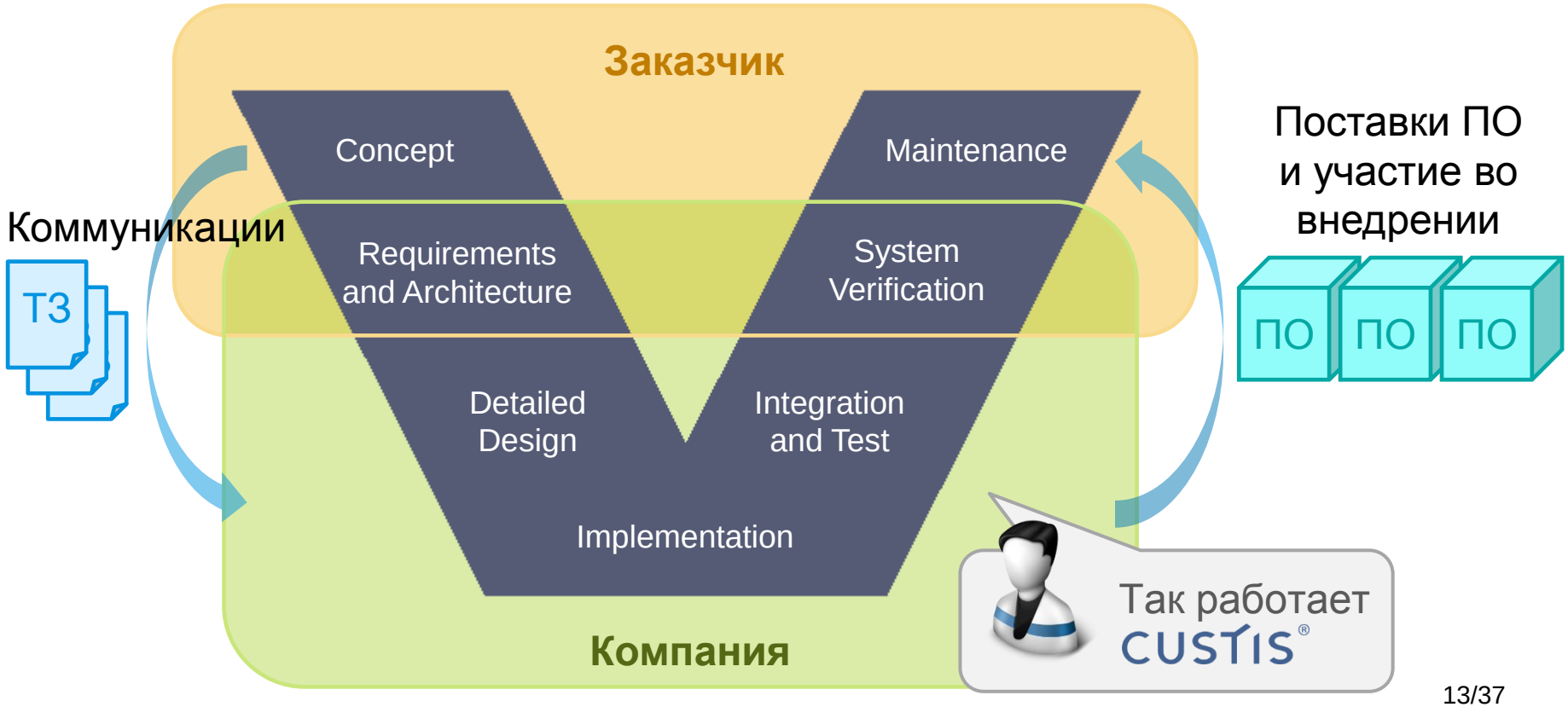
Jack W. Reeves «What is software design» (1992; перевод)

CUSTIS®

Решение – коммуникация

Это фишка **Agile**

Заказчик ставит задачу в терминах бизнеса, компания осуществляет разработку и внедрение вплоть до перестройки бизнес-процессов. Большая зона совместной ответственности обеспечивает успех проекта

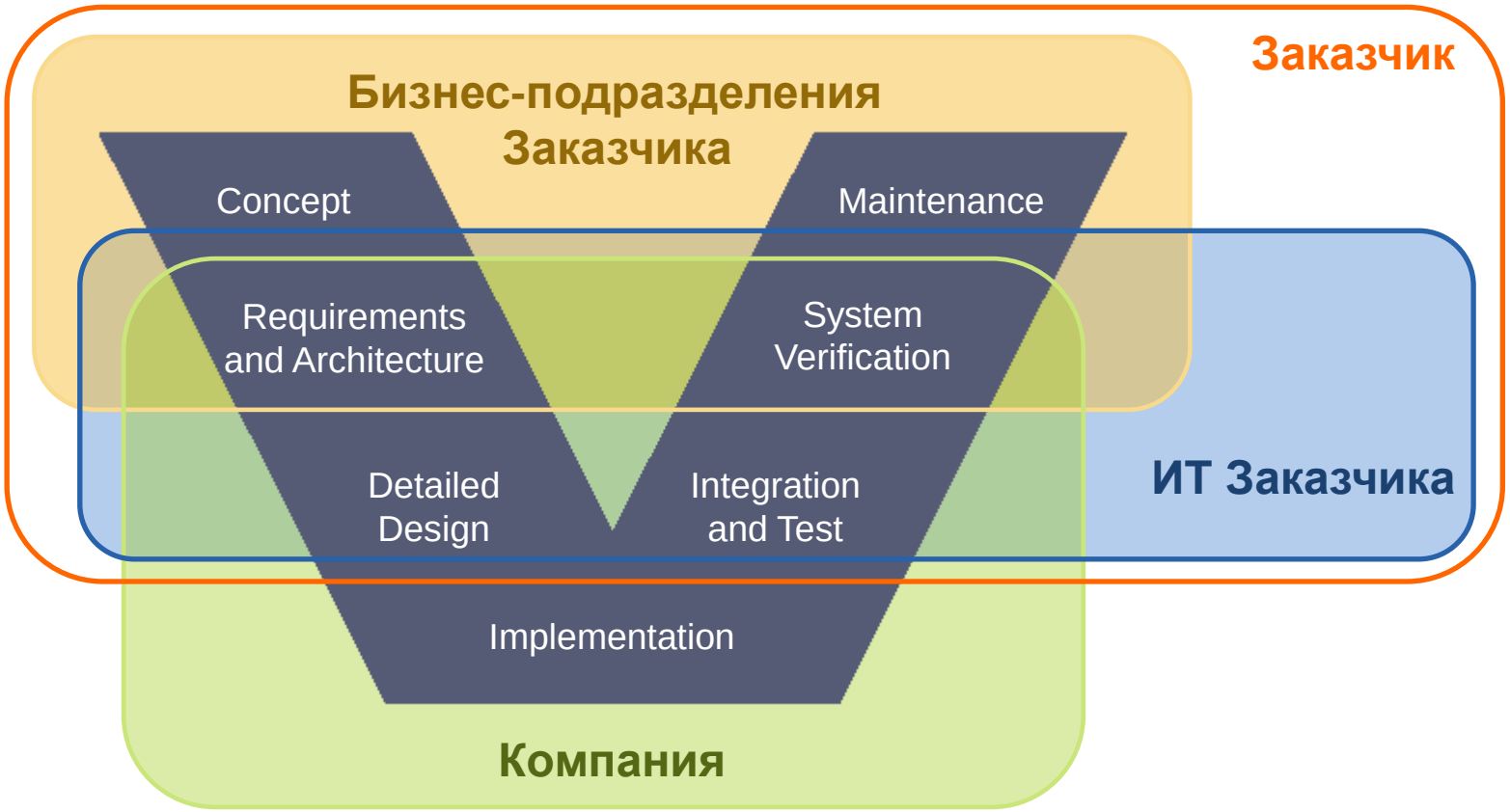


CUSTIS®

Заказчик и компания-разработчик: разделение ответственности и взаимодействие

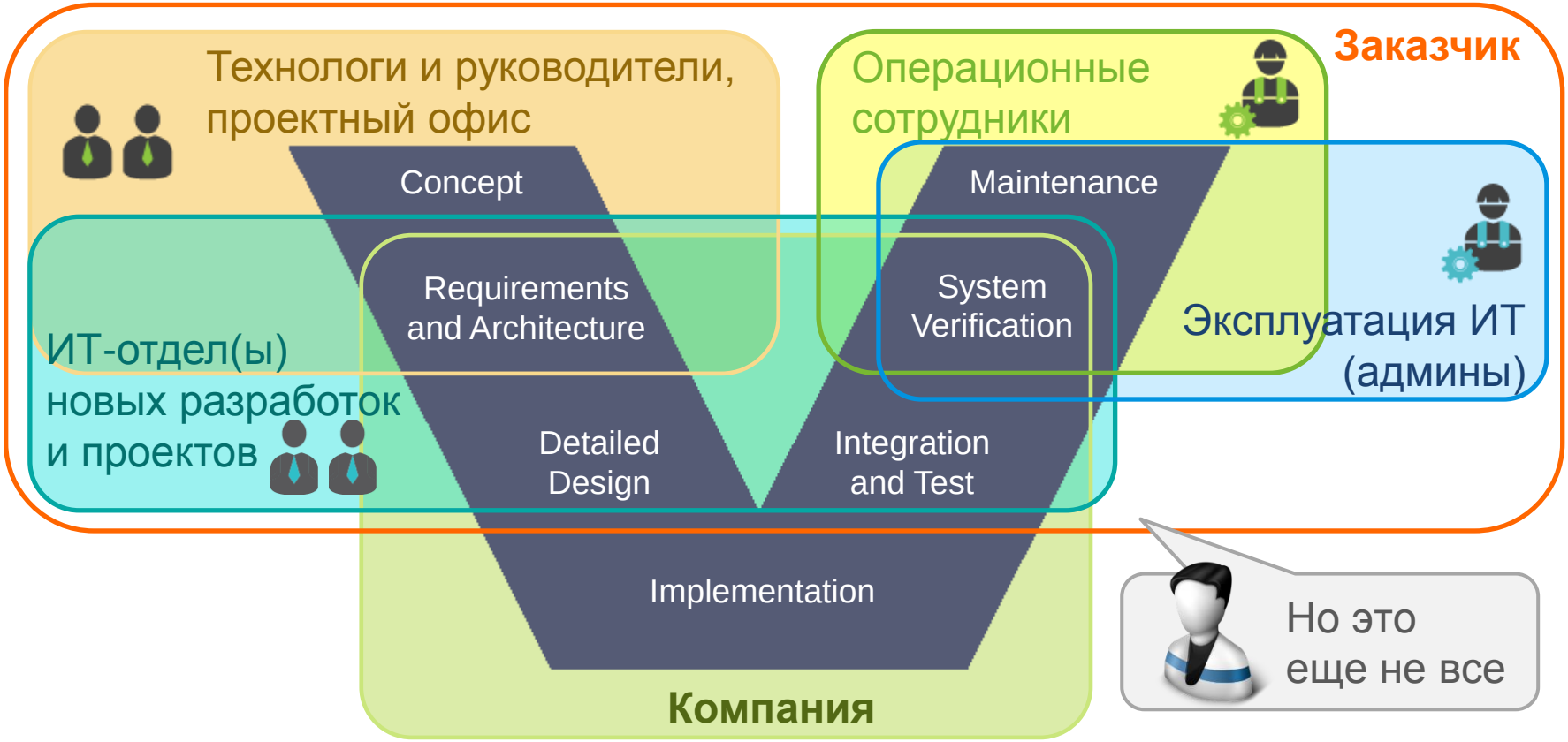
У заказчика есть свой ИТ…

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



Операционная работа и развитие

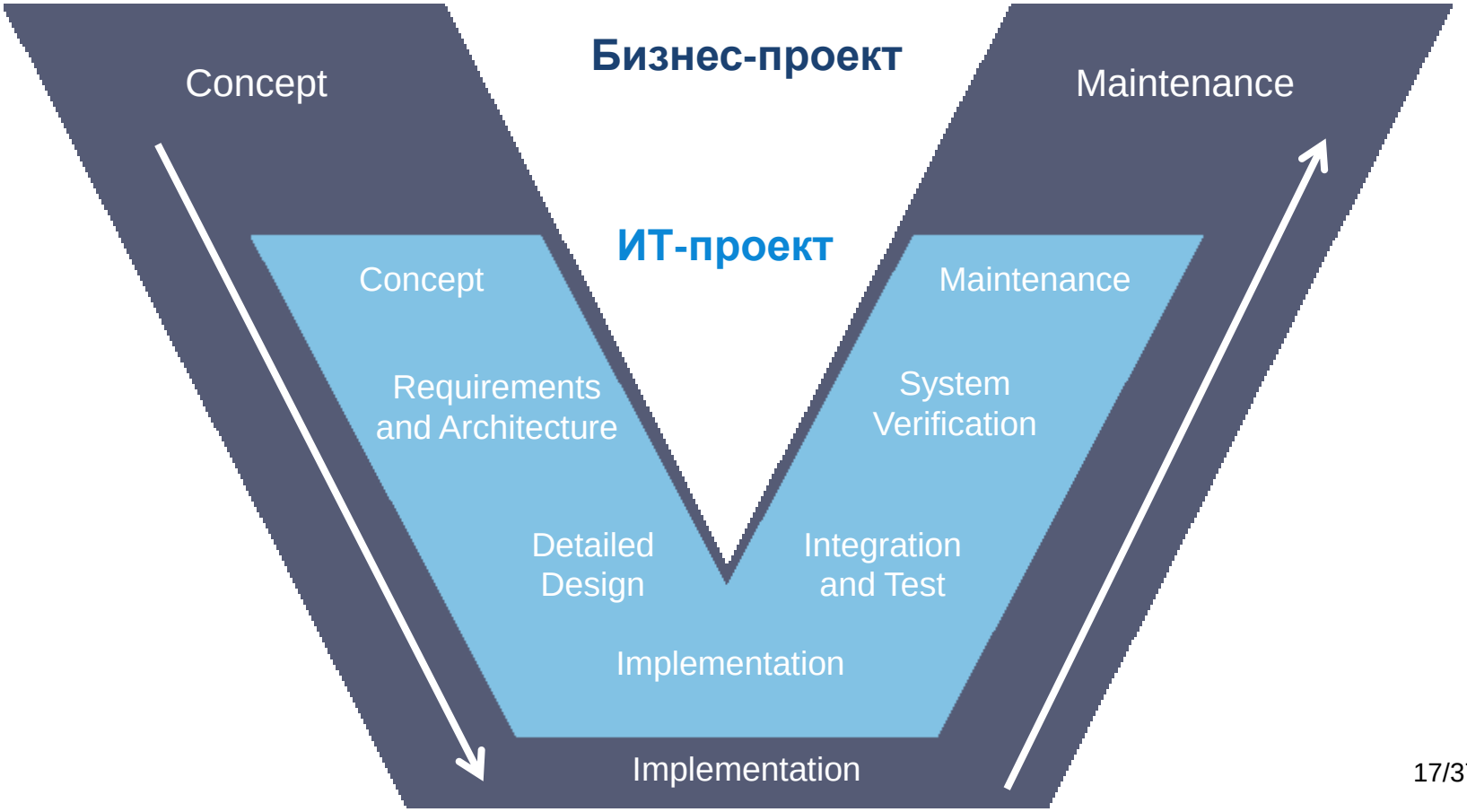
Постановку задачи на разработку и эксплуатацию созданного приложения осуществляют разные группы стейкхолдеров заказчика



CUSTIS®

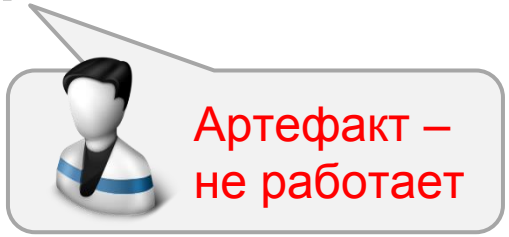
ИТ-проект – часть бизнес-проекта

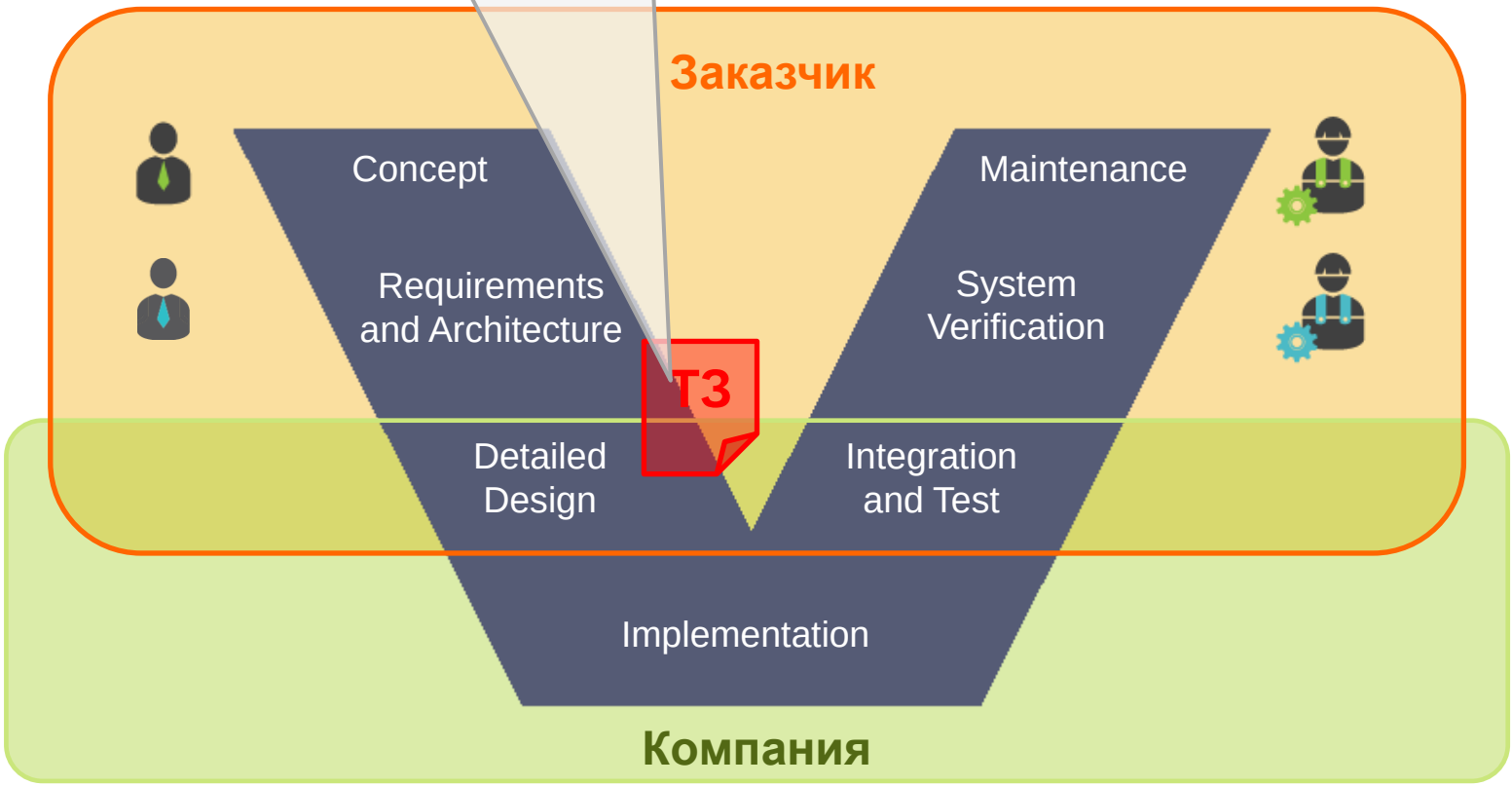
Ответственность стейкхолдеров заказчика – относительно бизнес-проекта, а не ИТ-проекта



Где ответственность за целое?

Процессный ответ – обеспечим через согласование артефакта

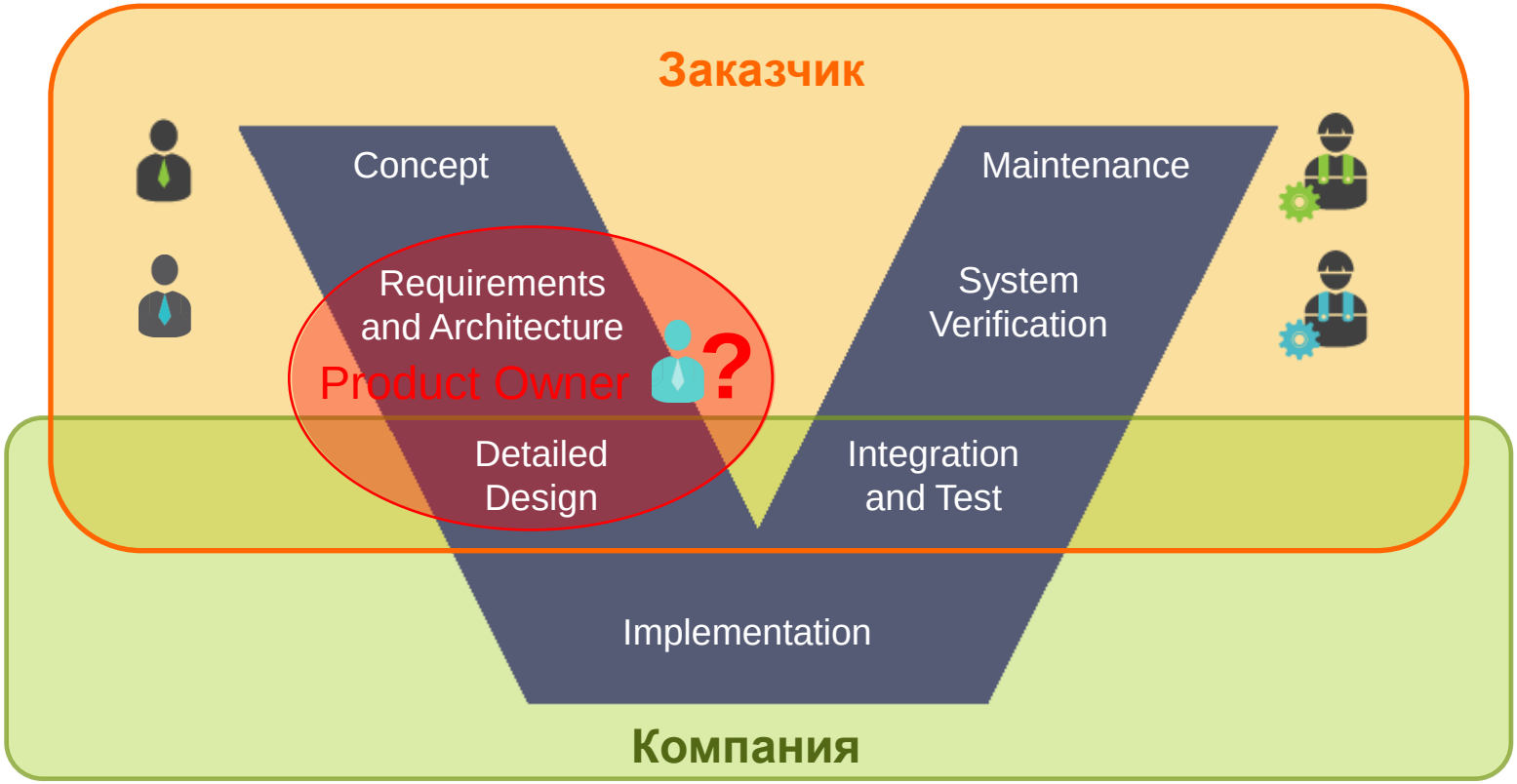




CUSTIS®

А пусть будет Product Owner

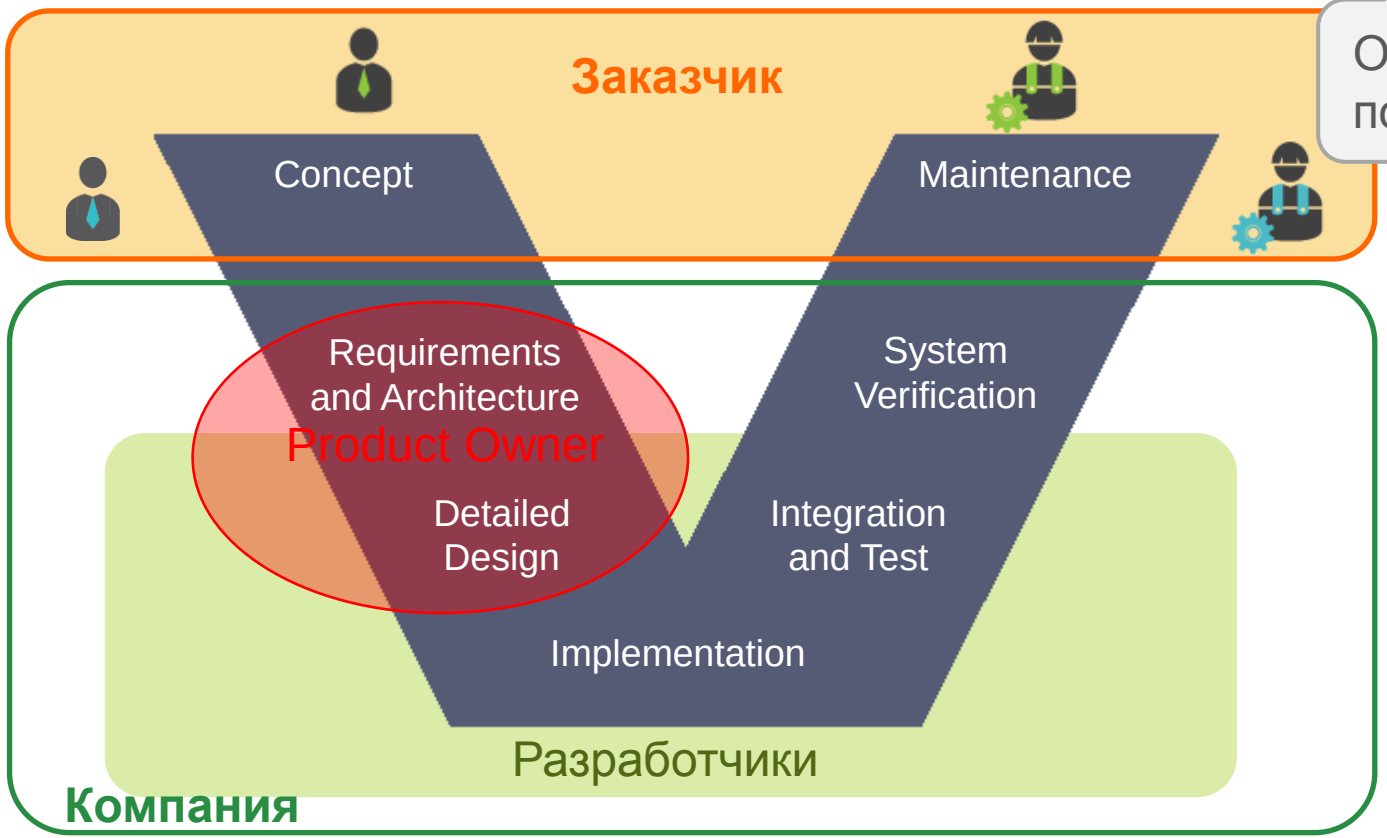
Только у заказчика
он не всегда есть



CUSTIS®

Заказчик часто не готов к такой области ответственности

Ответственность компании надо расширять,
частью ее становится **Product Owner,** он же **менеджер** ☺



Об этом будет подробнее

CUSTIS®

Ответственность в компании-разработчике

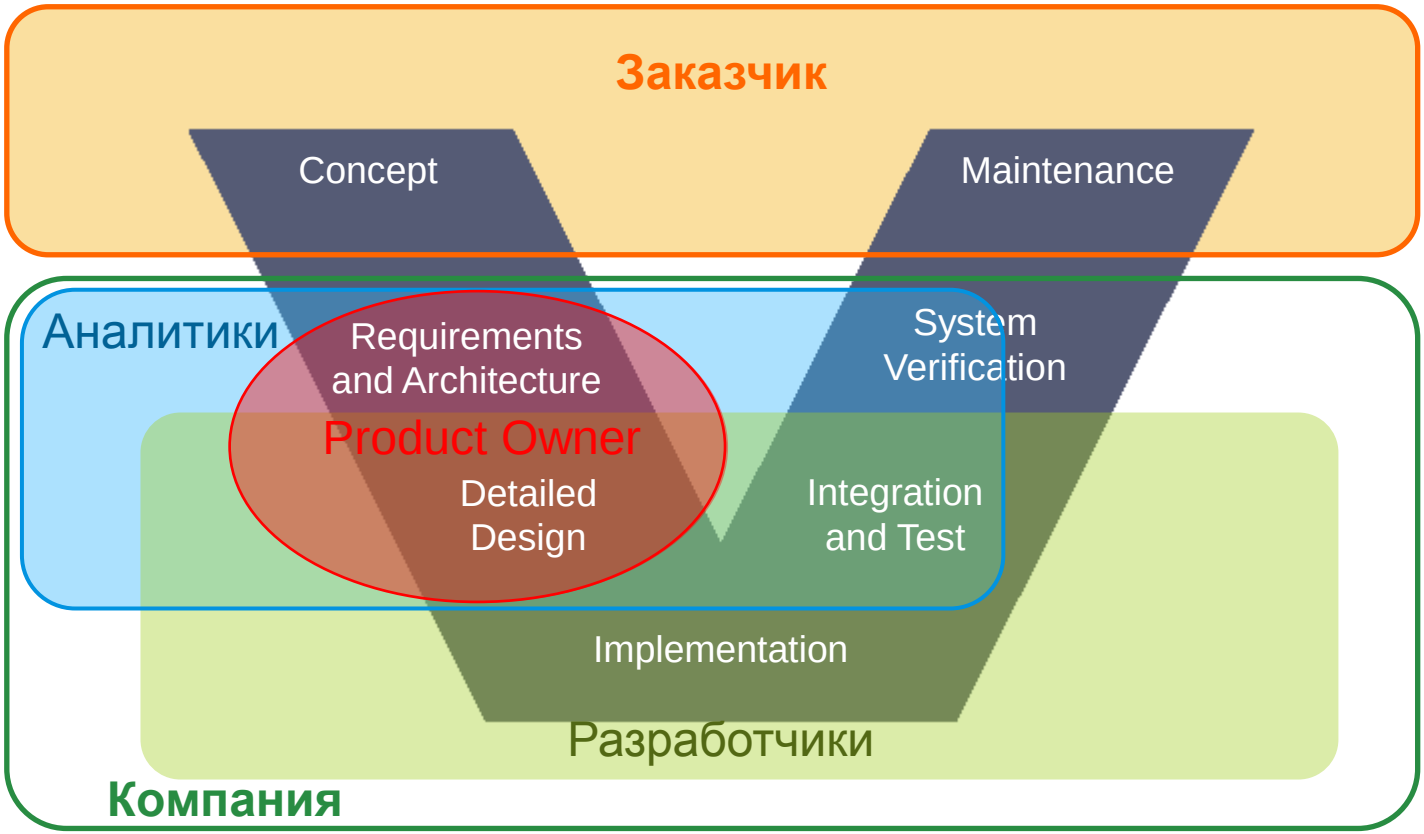
CUSTIS®

Аналитики тоже устраняют разрыв

Вопрос: Насколько аналитики заняты в тестировании и сдаче системы?

Вариант 1: Сдают разработчики

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



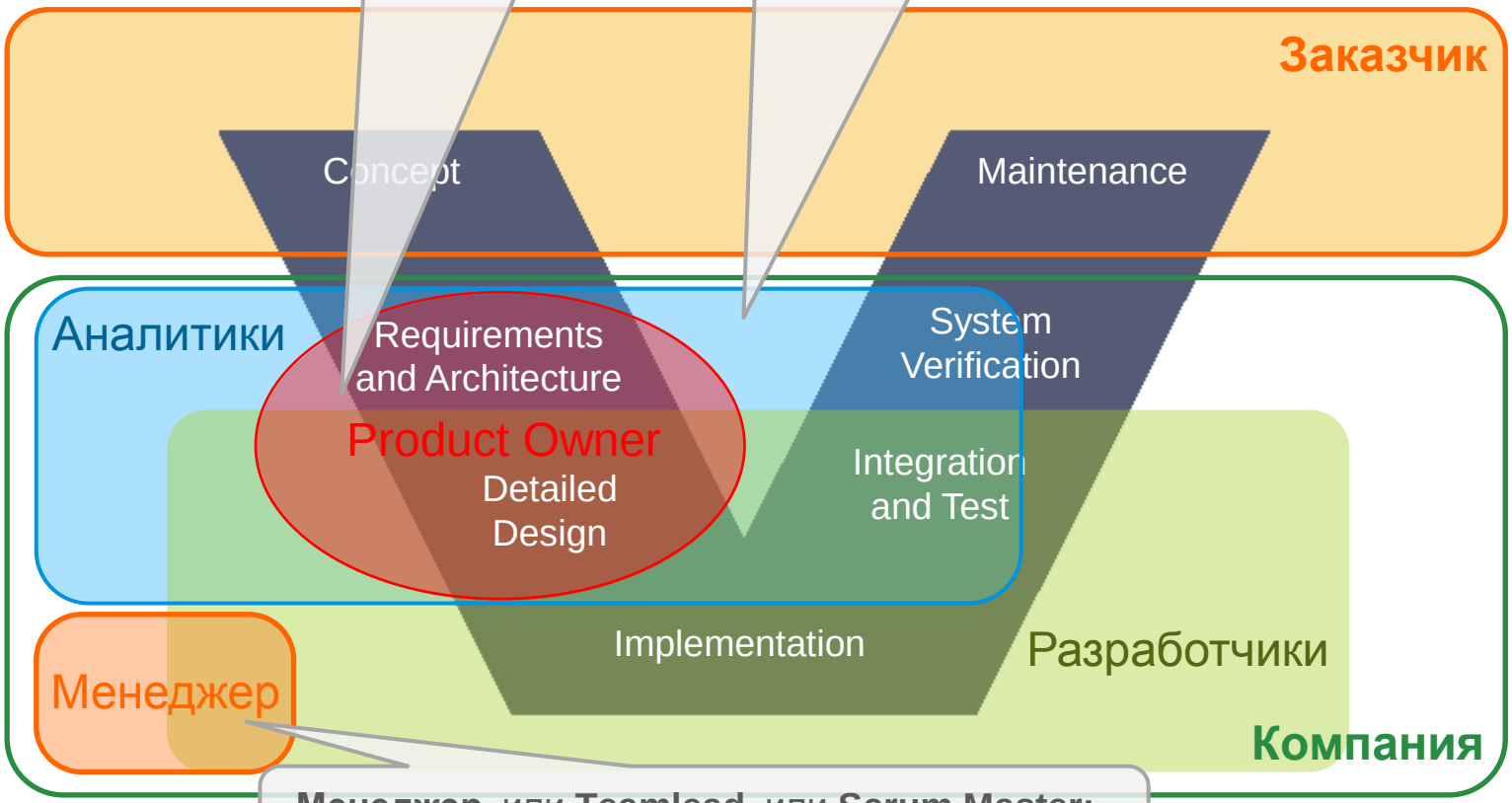
CUSTIS®

Менеджер, Product Owner, аналитик

Product Owner: управление целостностью и порядком разработки системы

Аналитик: преобразование требований в модель системы

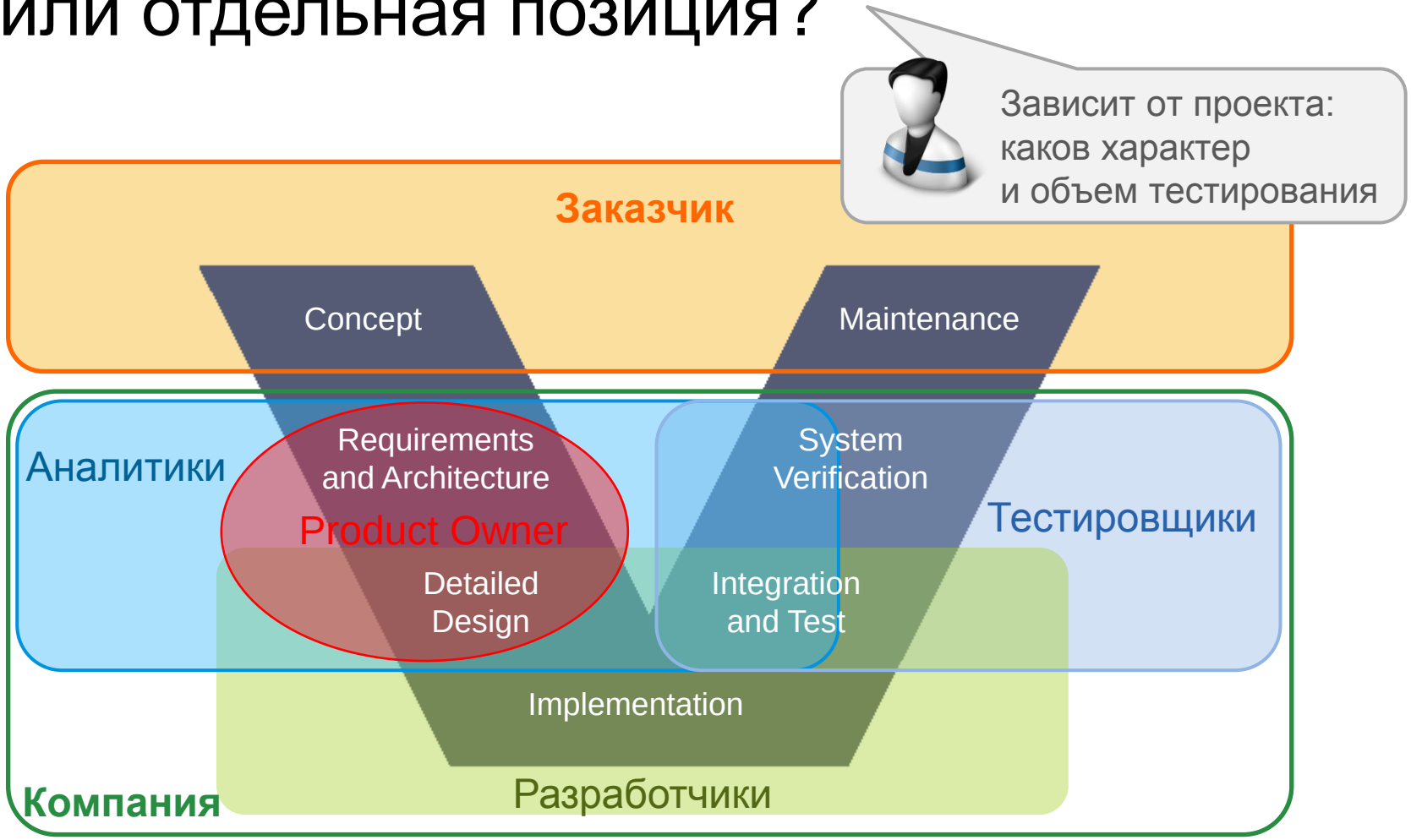
Роли, а не должности



Менеджер, или **Teamlead,** или **Scrum Master:** организация процесса работ и управление им

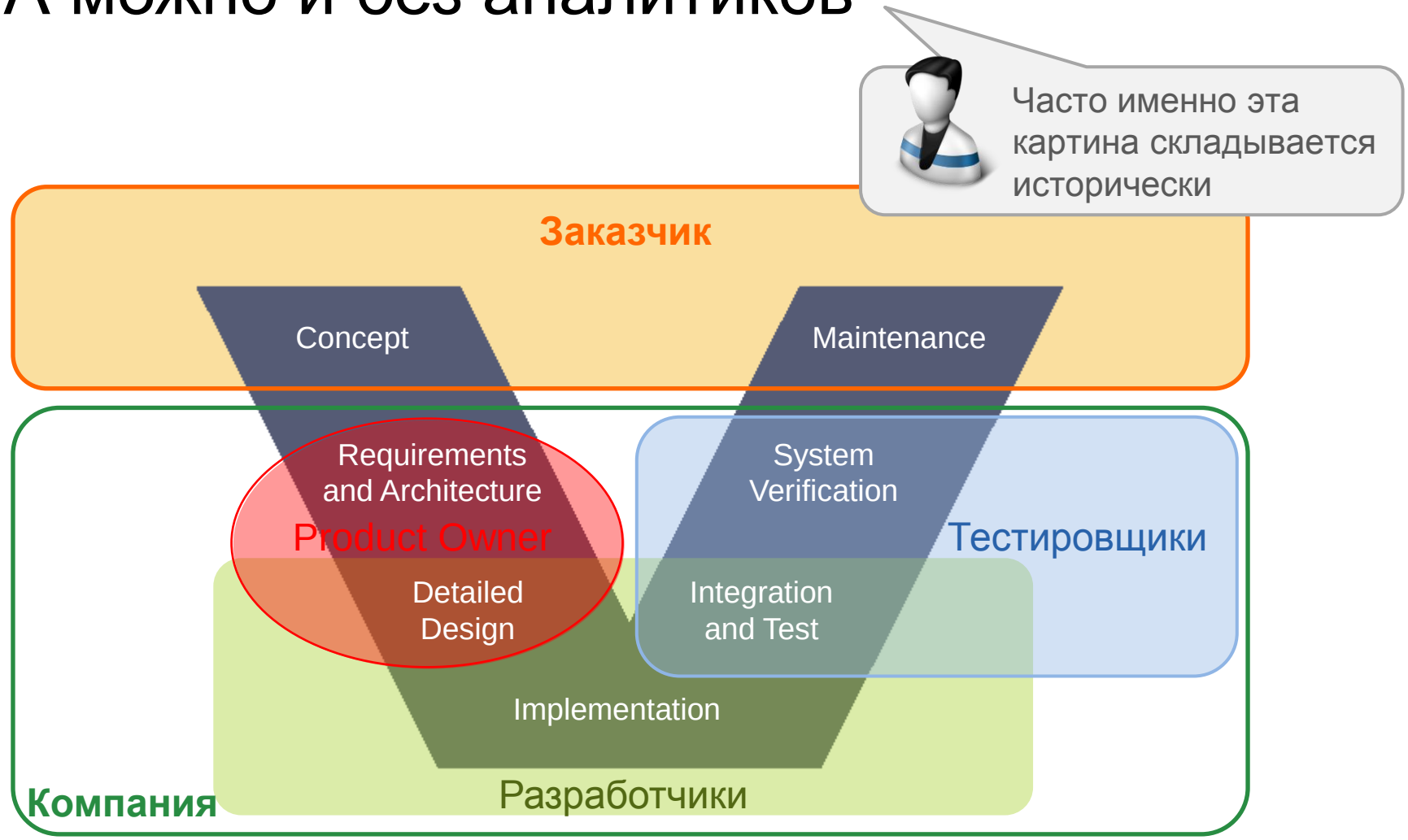
CUSTIS®

Тестировщик: младший аналитик или отдельная позиция?



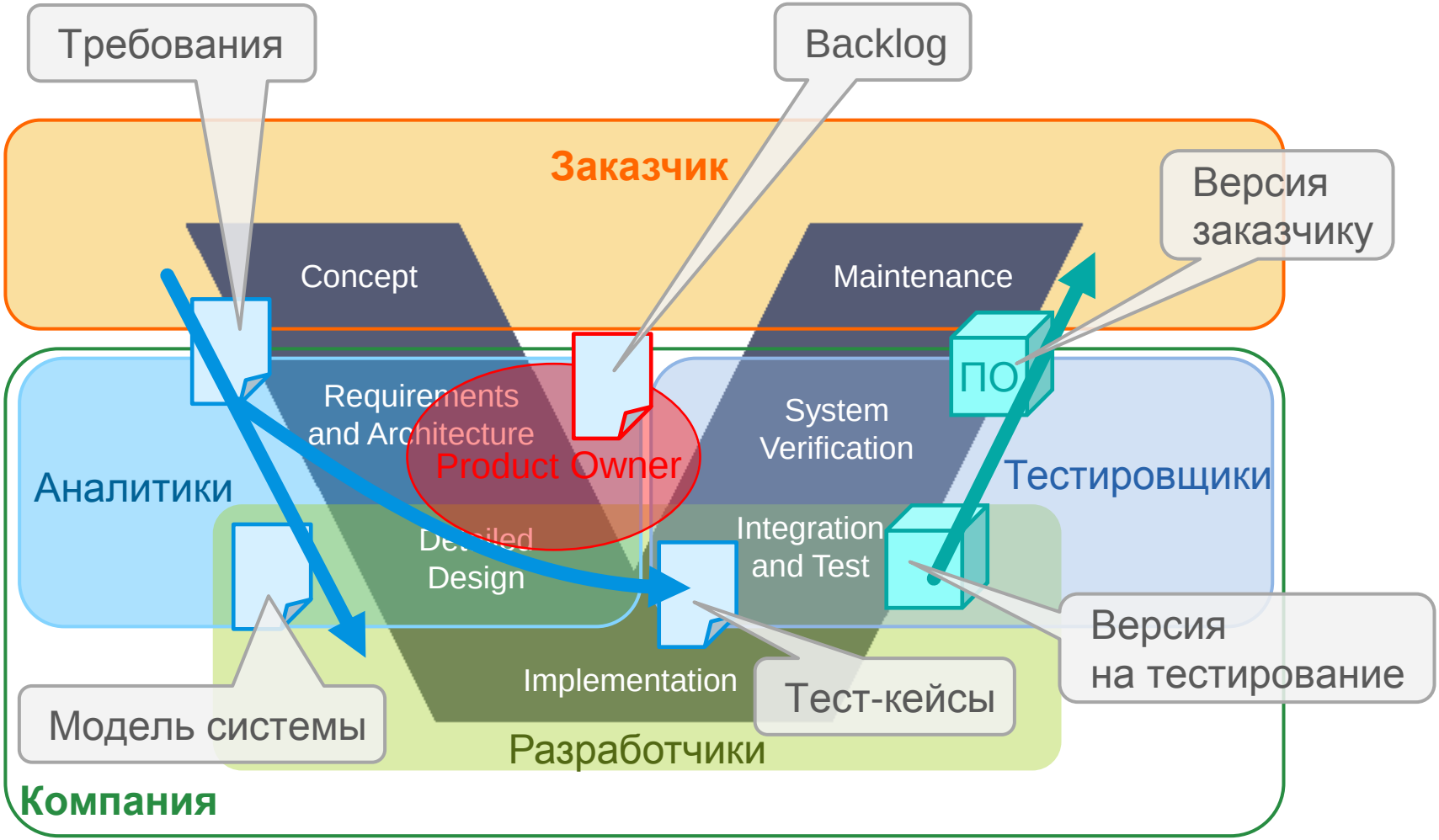
CUSTIS®

А можно и без аналитиков



CUSTIS®

Артефакты на проекте

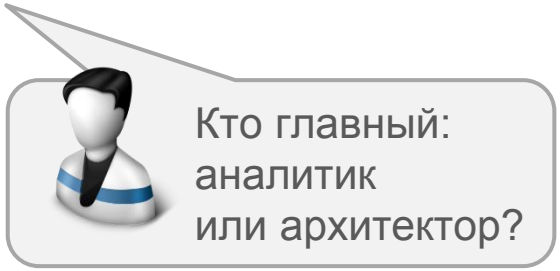


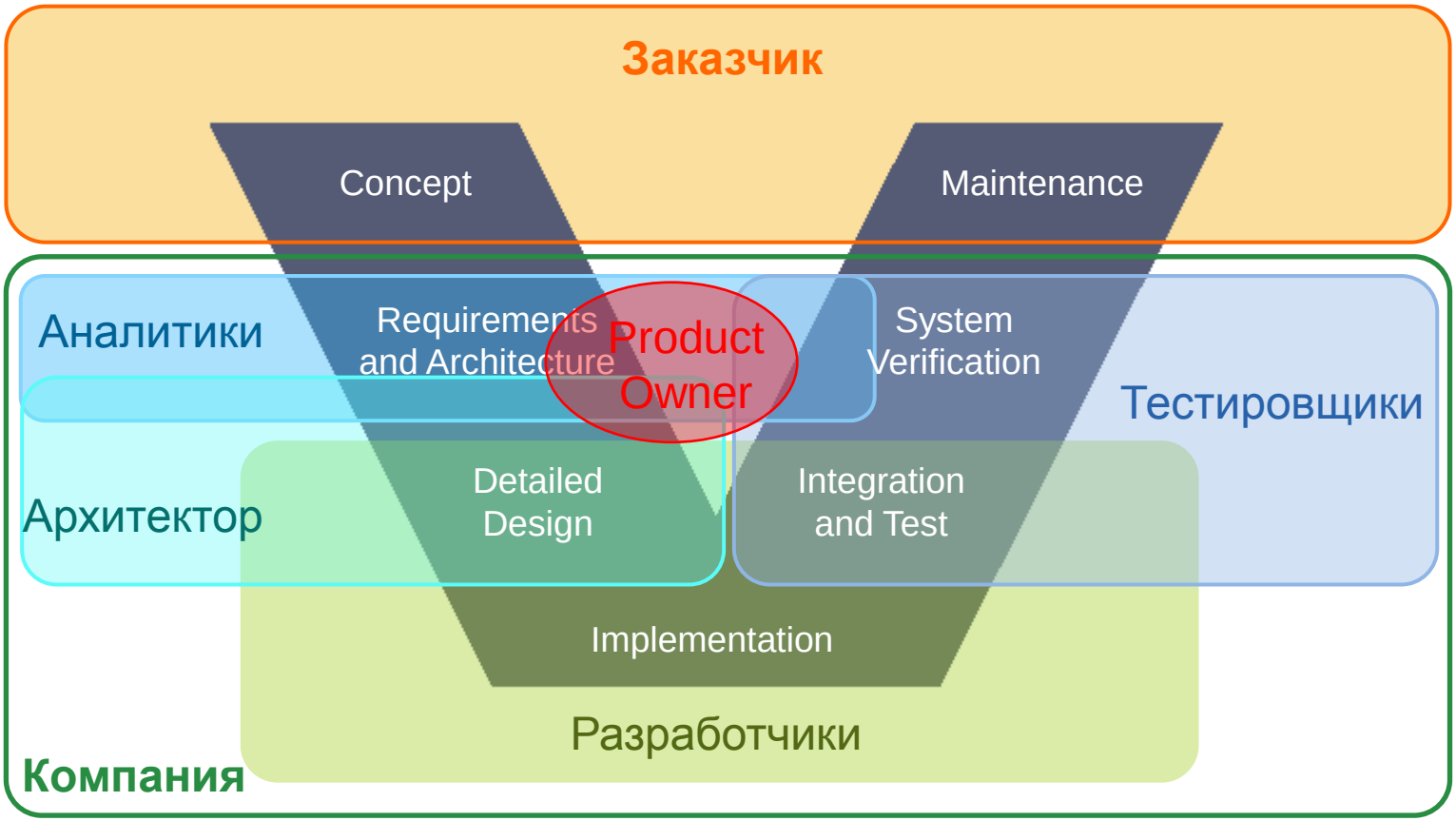
CUSTIS®

Архитектор, он же Techlead

1. Построение начальной архитектуры
2. Проектирование типовых решений

   Это – разные задачи **и позиции**

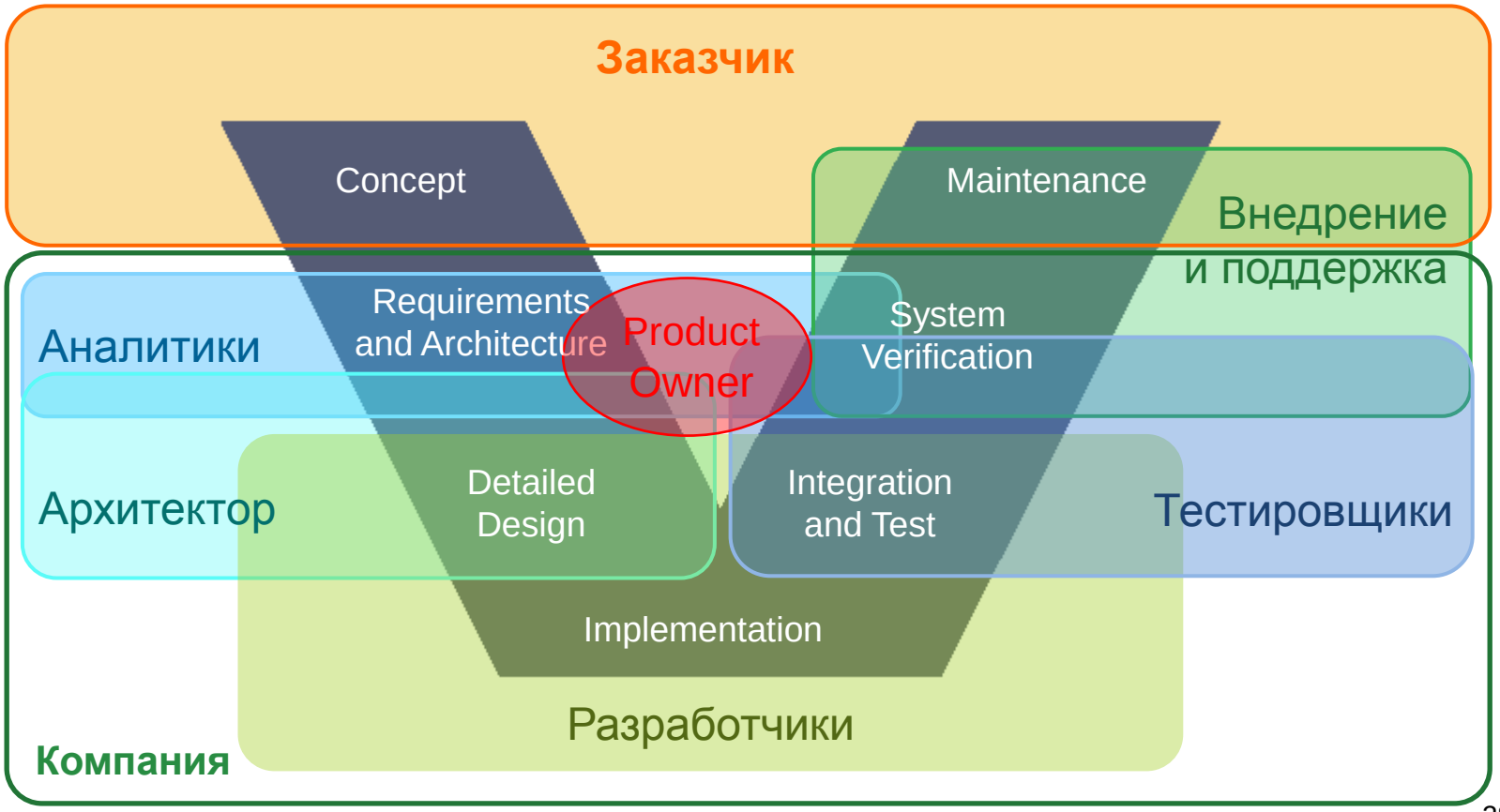




CUSTIS®

Еще есть внедрение и поддержка

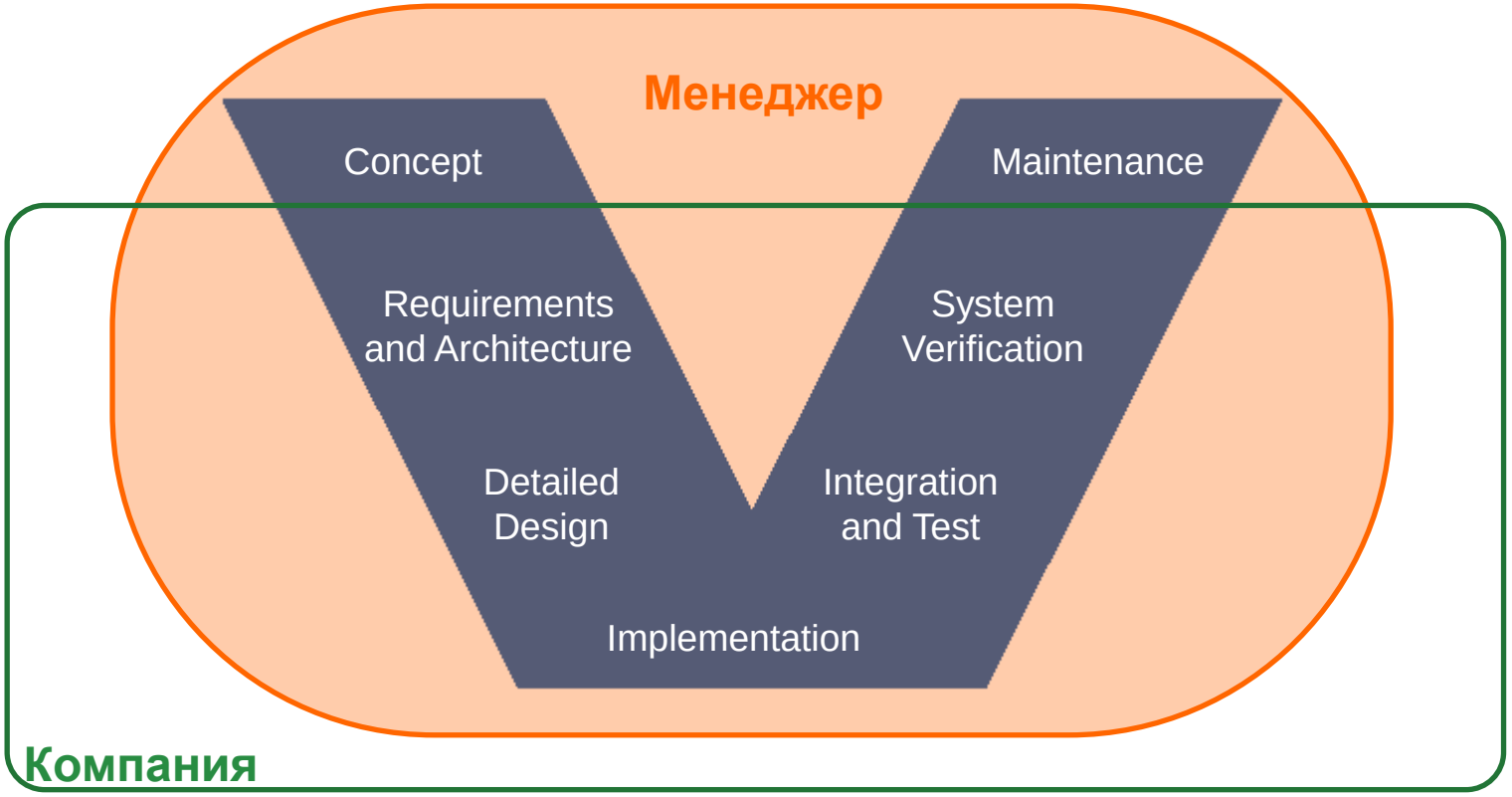
Разделение работ с заказчиком может быть различным
и часто – **неявным**



CUSTIS®

Простое управление: за все отвечает менеджер

Менеджер перед компанией может отвечать за весь процесс работ на проекте, включая область ответственности заказчика

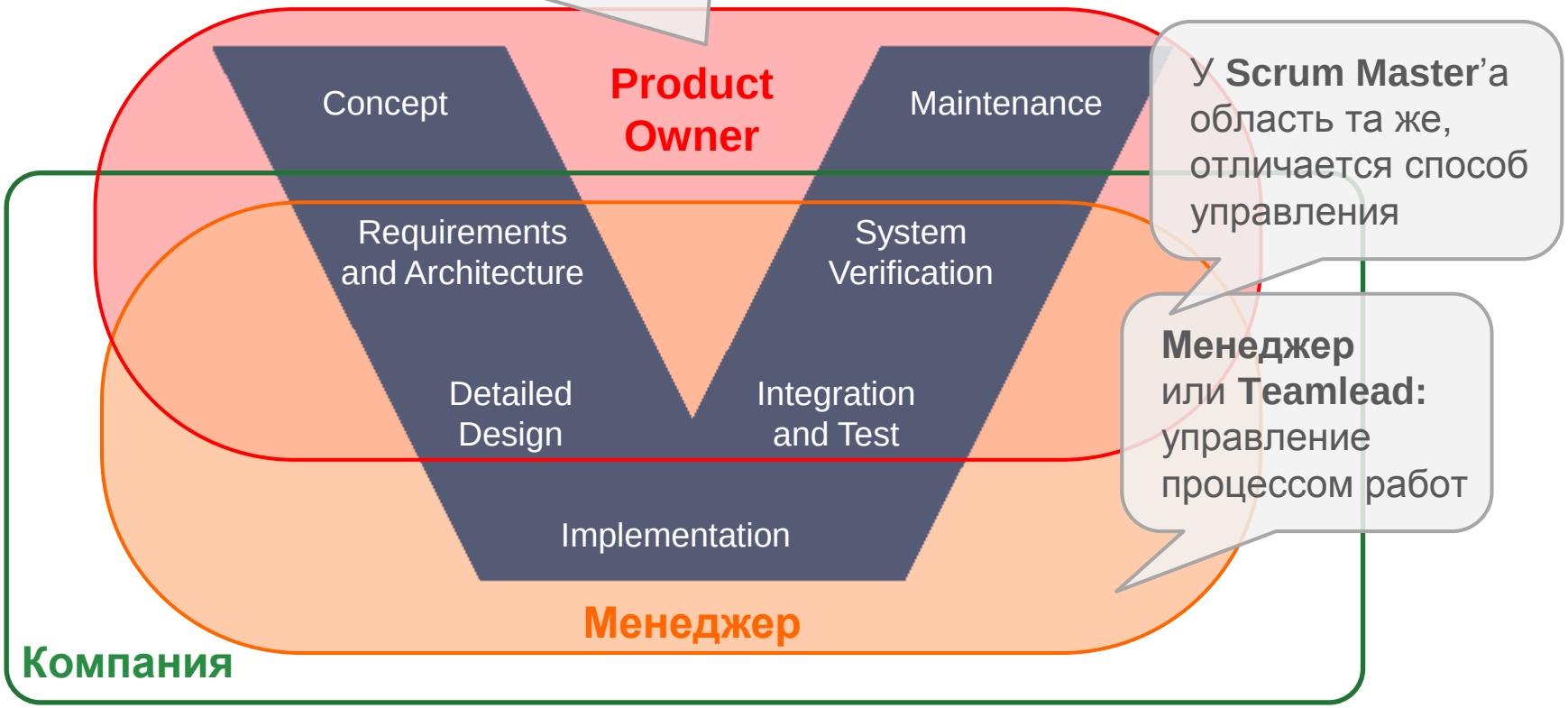


CUSTIS®

Разделение управления: менеджер и Product Owner

На V-модели разделение не видно – меняем схему

Product Owner: управление целостностью и порядком разработки системы

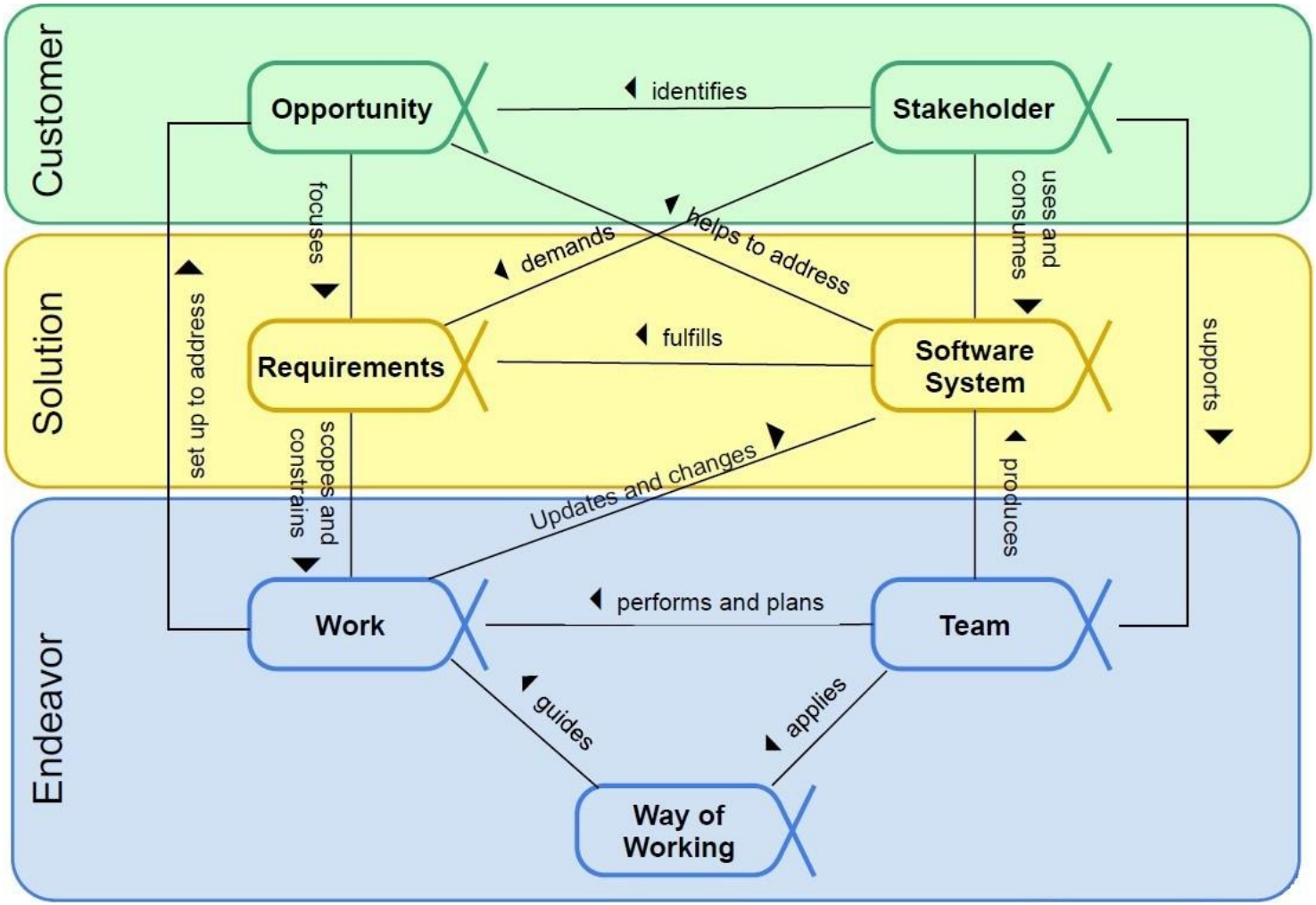


У **Scrum Master**'а область та же, отличается способ управления

Менеджер или **Teamlead:** управление процессом работ

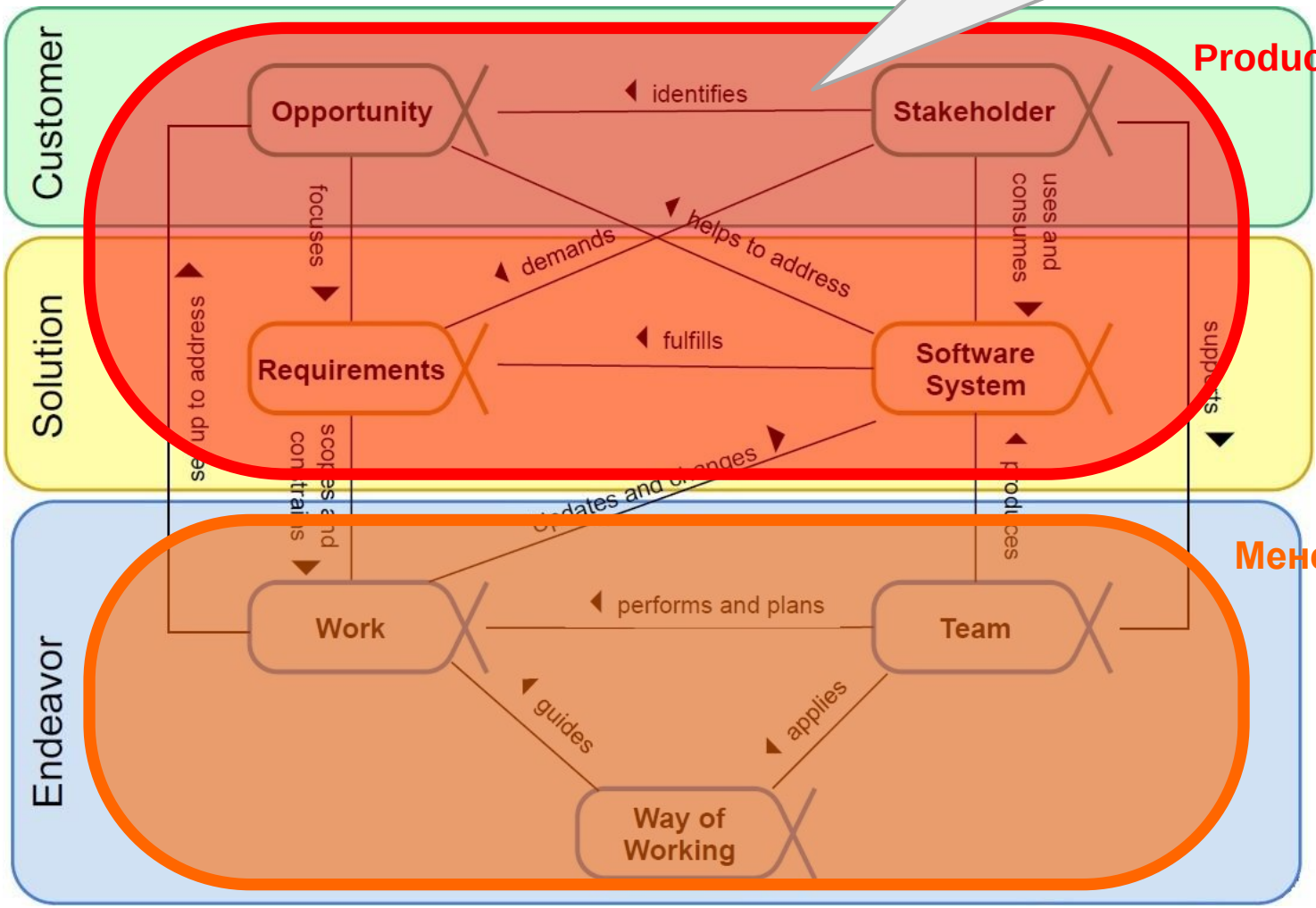
CUSTIS®

OMG Essence: объекты процесса



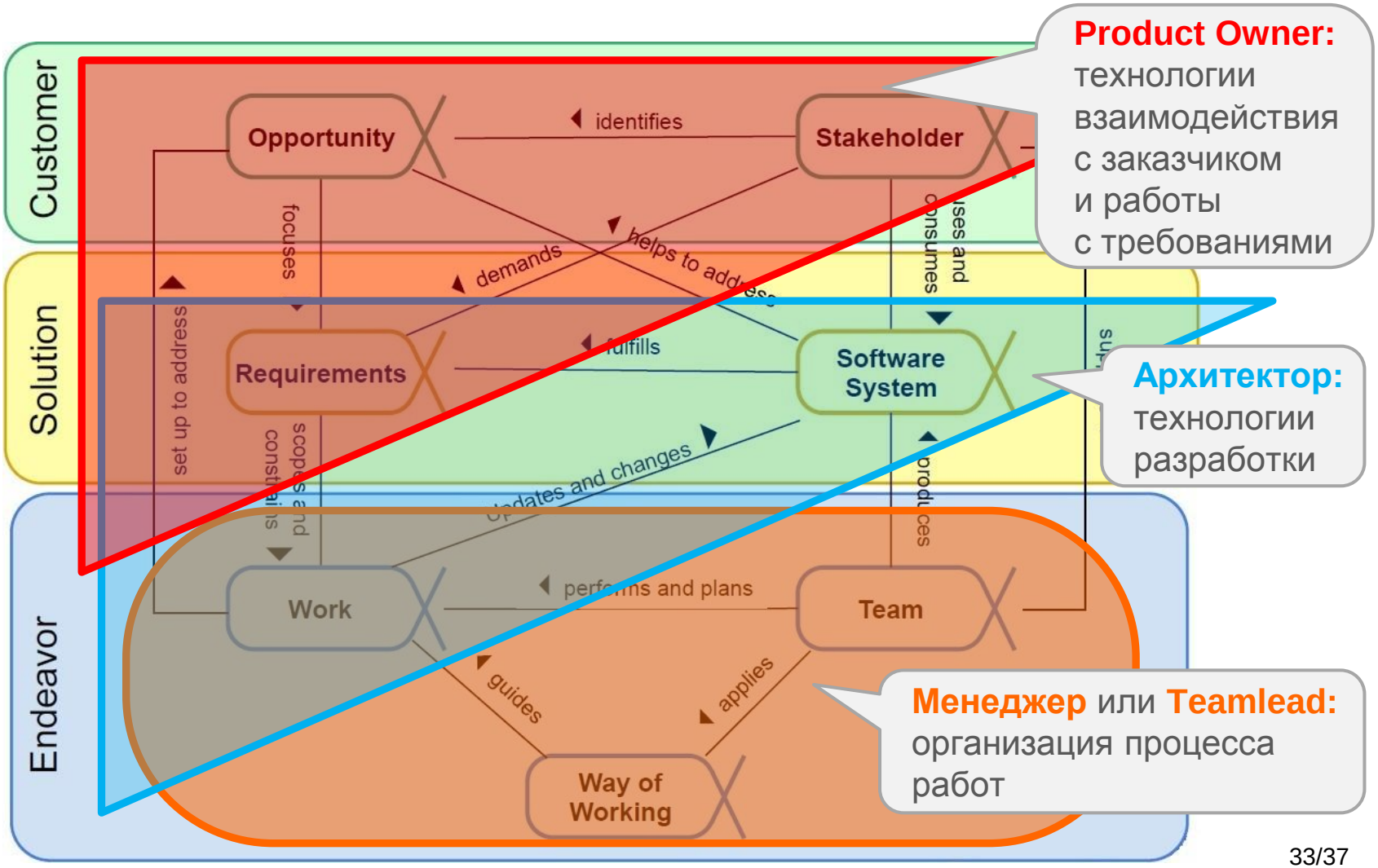
Области управления

Разделение облегчило поиск управленческого персонала и **обеспечило успех Agile**



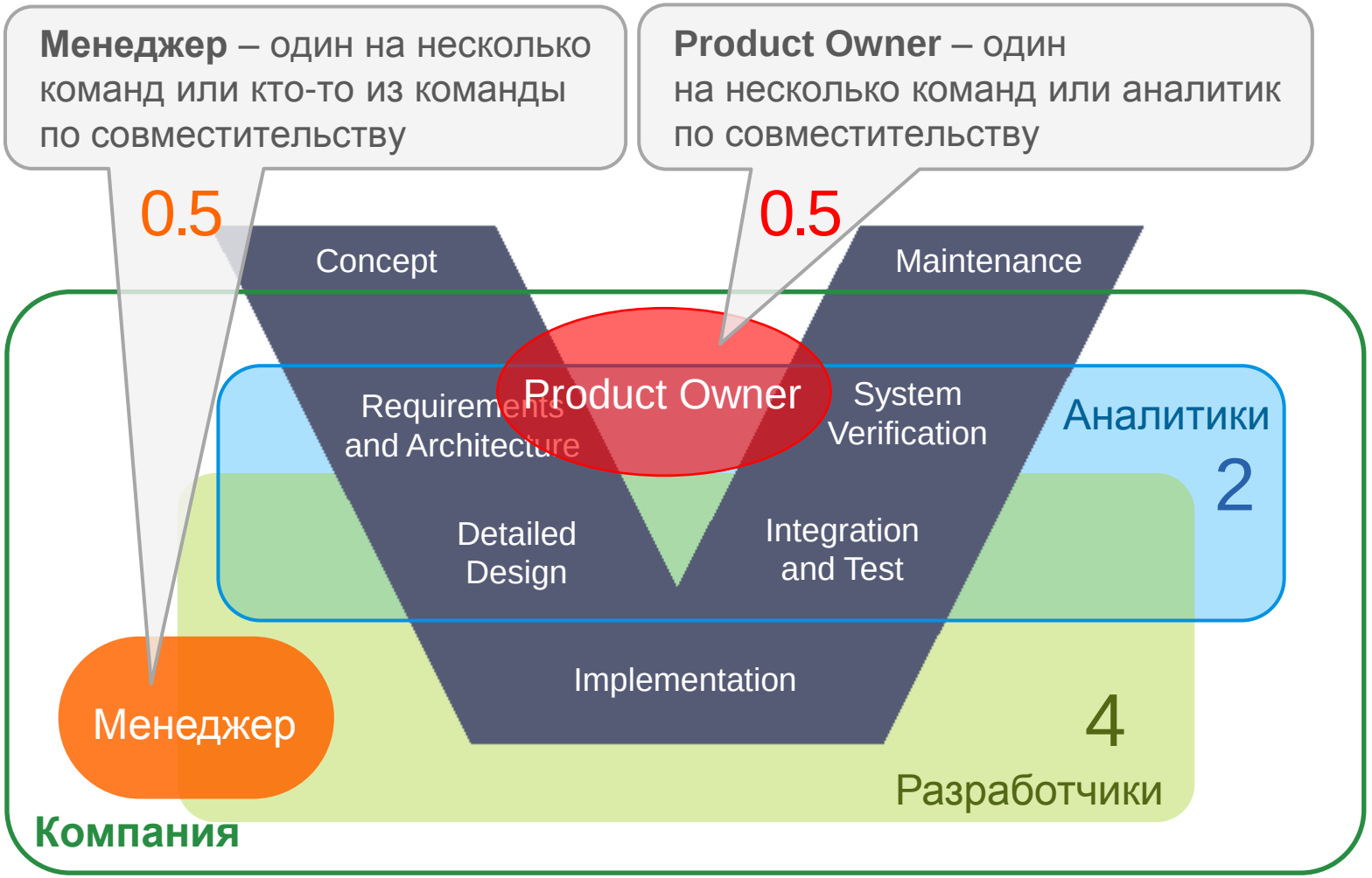
CUSTIS®

Области технологизации



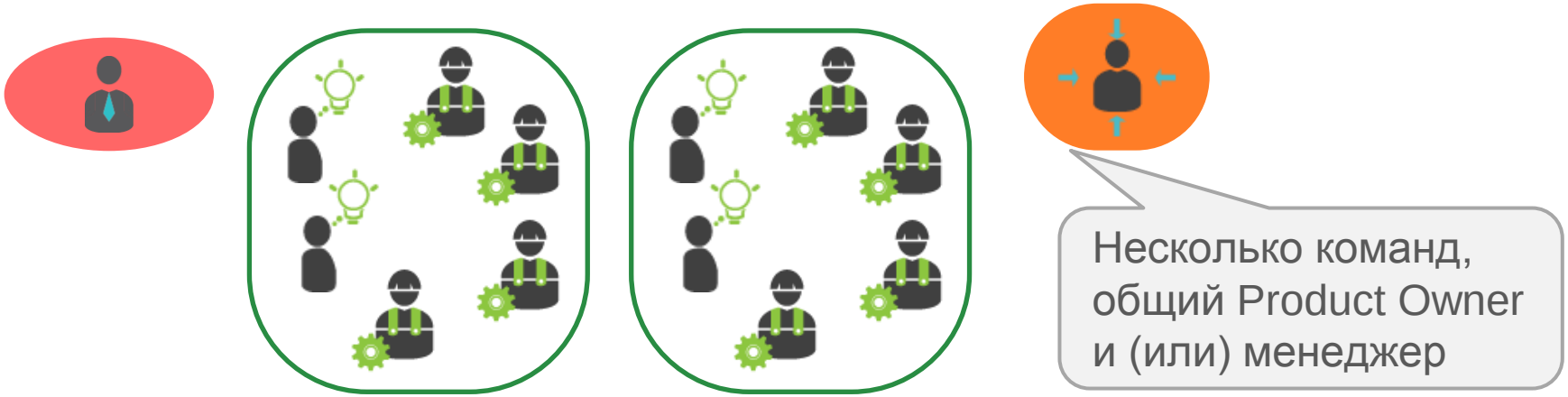
CUSTIS®

Размер команды – 7 ± 2



CUSTIS®

А если проект большой?



CUSTIS®

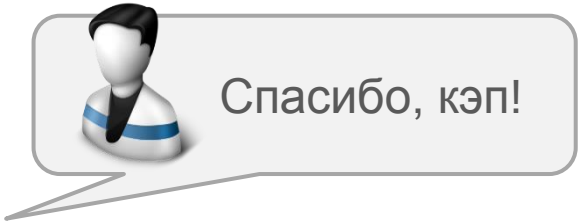
А если проекты маленькие?

- Если проекты достаточно однородны, можно передавать одной команде без специализации
- Можно делать мини-группы по каждому проекту
- Решение зависит от равномерности потока работ по проектам

CUSTIS®

Подводя итоги

- Разделение ответственности зависит от взаимоотношений с заказчиком
- От характера вашего проекта
- И от традиций компании



Не существует общепринятого или единственно правильного способа

Его надо проектировать!

Спасибо! Вопросы?

Максим Цепков mtsepkov.org