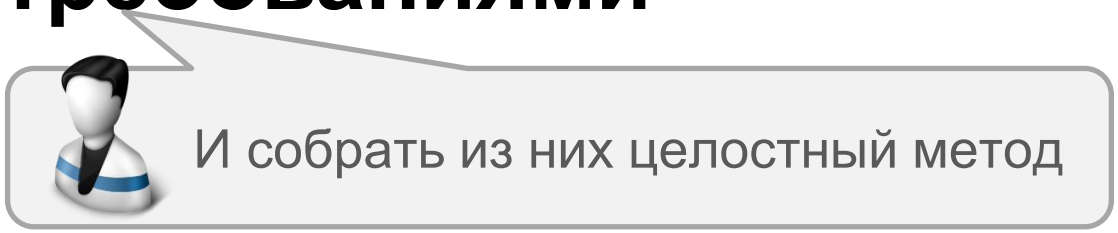
Analyst Days 2017

CUSTIS®

Как выбрать для проекта практики проектирования и работы с требованиями



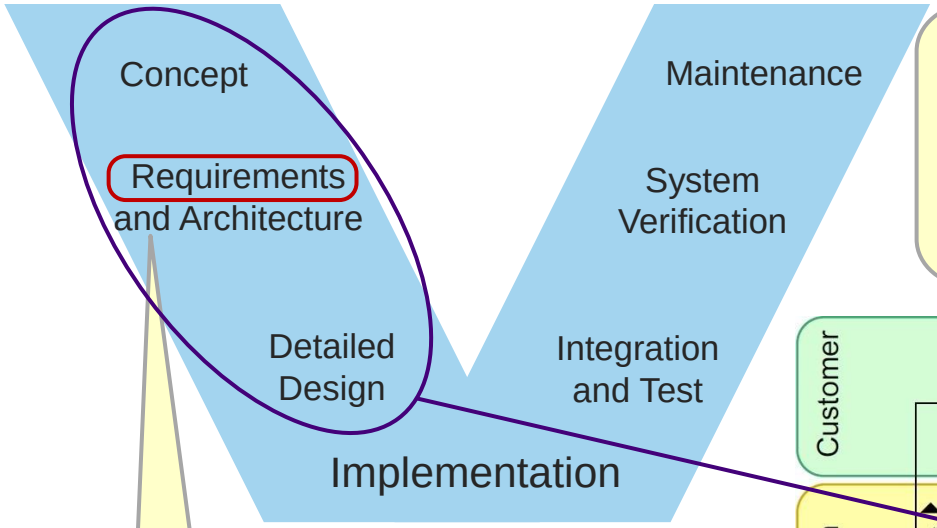
Максим Цепков

Главный архитектор дирекции развития решений

Analyst Days

Москва, 21–22 апреля 2017

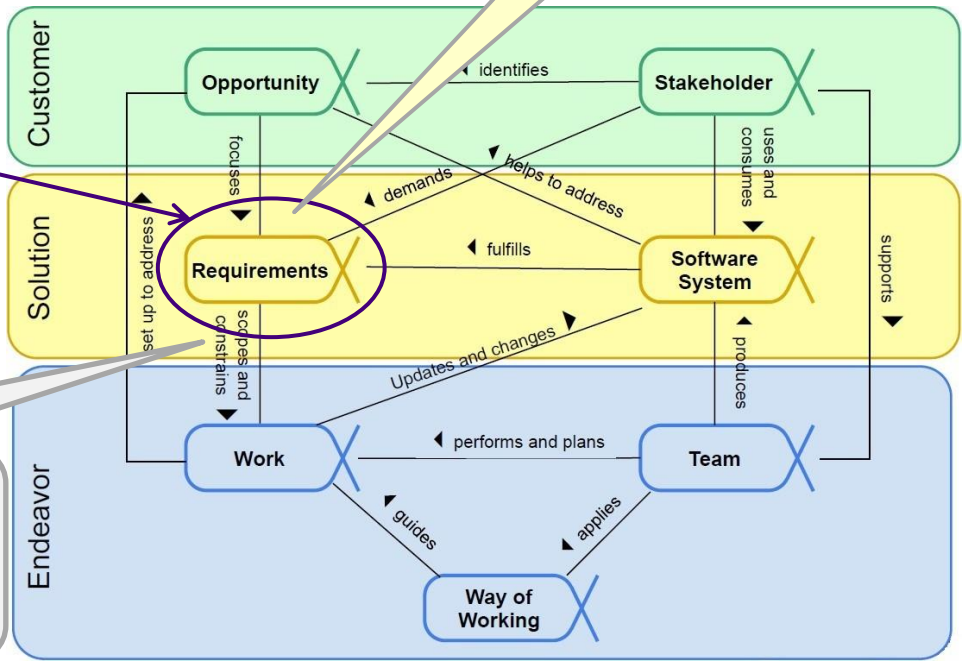
Что такое «требования»: два ответа



В OMG Essence **требования – все описание системы** от потребностей пользователей до детального дизайна. Возможности – отдельно

На V-диаграмме (Wikipedia) **требования – внешние функции** системы

Я буду говорить о требованиях в широком понимании **OMG Essence**

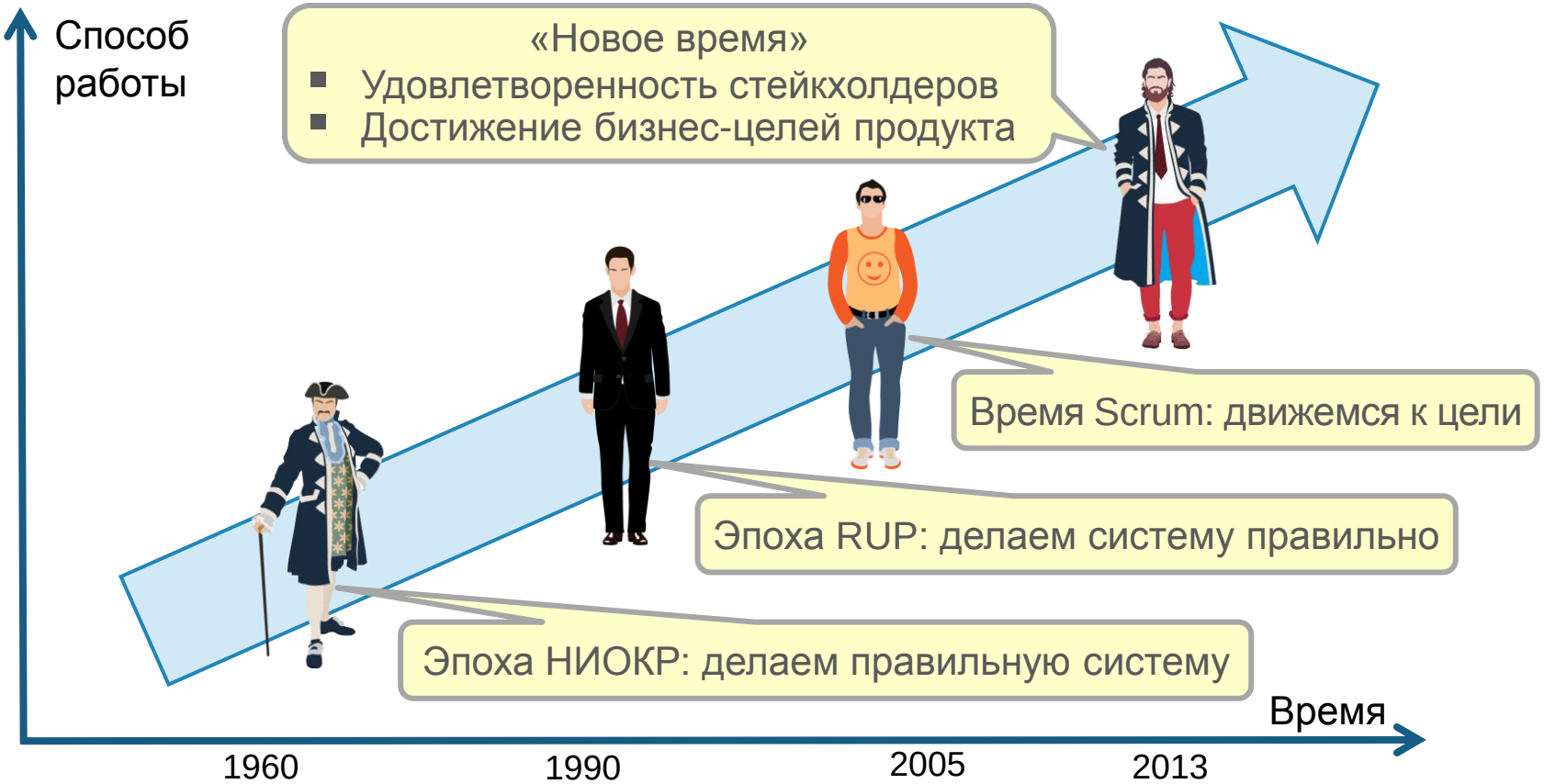


CUSTIS®

Надо ли собирать метод?

- Когда-то давно проектирование считалось искусством – как стиль художника
- Потом начали верить в **правильный** метод, который знают гуру
  - И когда Rational собрал трех методологов, Гради Буча, Ивара Якобсона и Джеймса Рамбо, у этой веры даже появились основания – родился UML. **Но не случилось**
- Потом пришло многообразие легких методов: проекты **разные,** и делать их надо по-разному

Big Picture развития методов



Подробности – в докладах «Развитие критериев качества и управления проектами» на AgileDays – 2015 и «Ответственность за качество в разных ИТ-проектах» на SQA Days – 20

CUSTIS®

О чем будет доклад

- Я не буду делать обзор методов – о каждом можно прочитать
- Я расскажу о вопросах, с помощью которых можно выбрать метод или собрать свой
- Речь пойдет о работе с требованиями, а бизнес-анализ, проектирование и смежные области будут только затрагиваться

Метод для работы с требованиями правильно собирать аналитикам: каждый сам кузнец своего счастья ☺

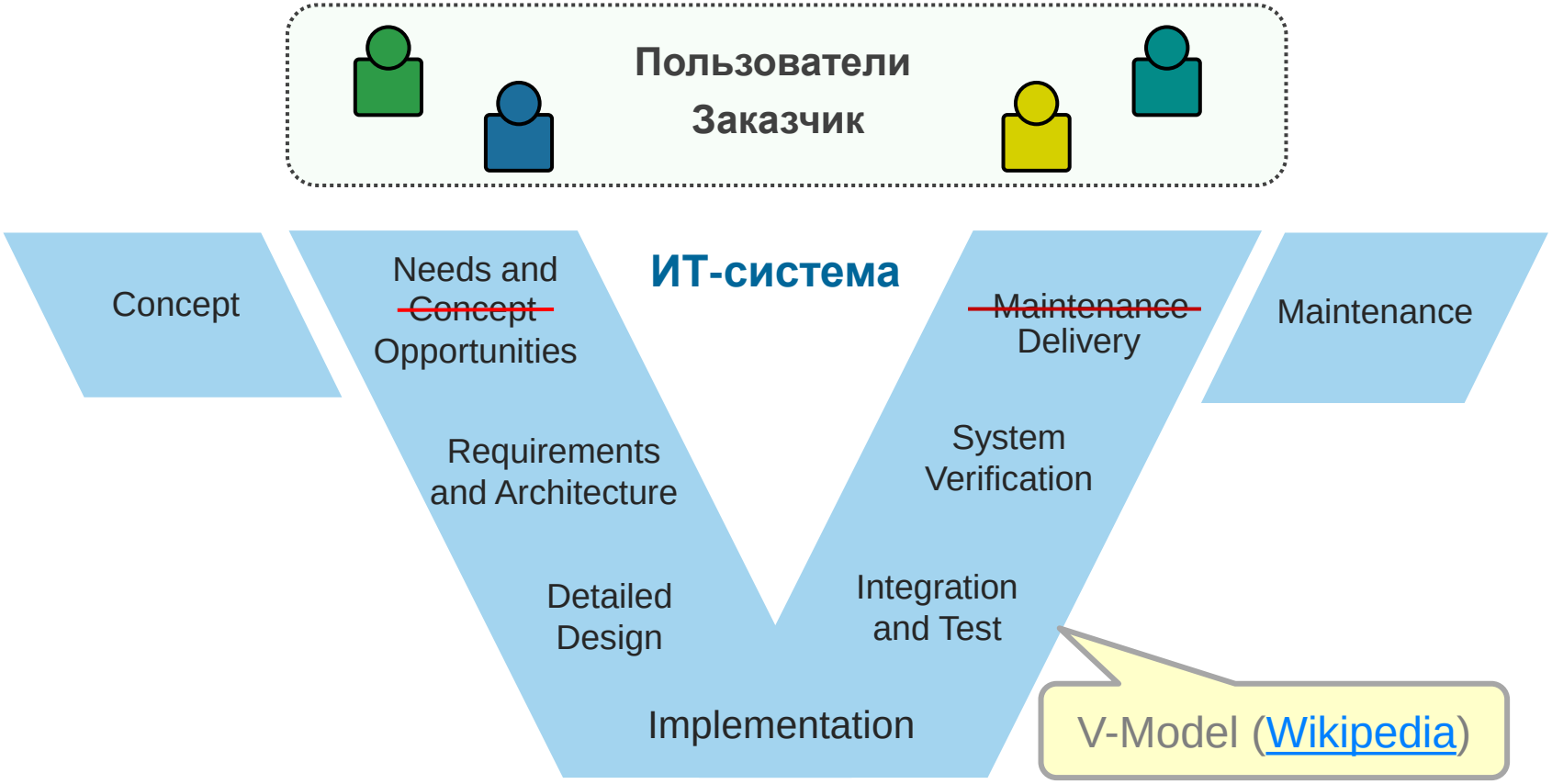
CUSTIS®

План рассказа

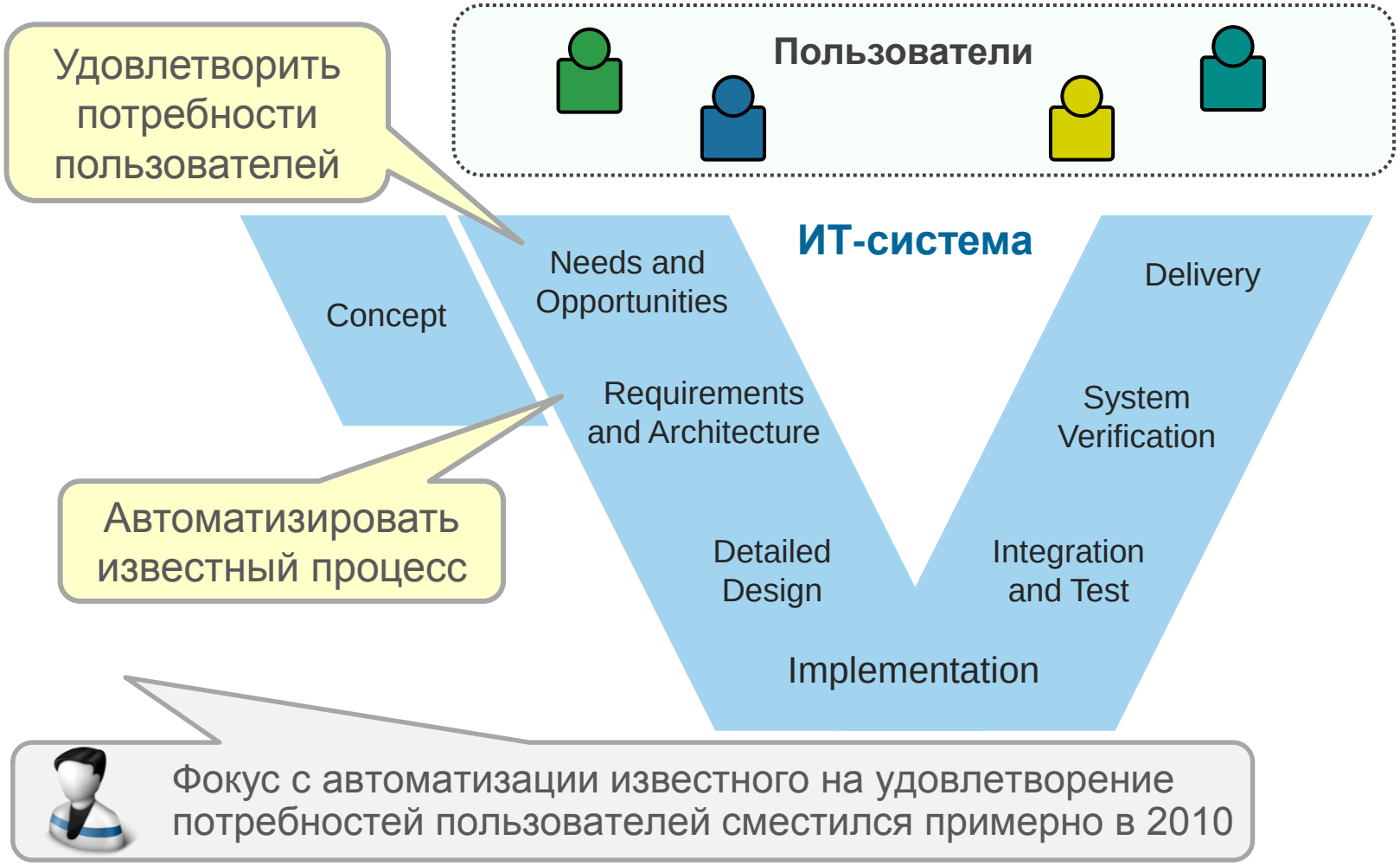
- Как определяем границы проекта?
- Как декомпозируем и инкрементально развиваем?
- Как обеспечиваем качество?
- Как организуем артефакты?
- OMG Essence – способ описать метод

Как определяем границы проекта?

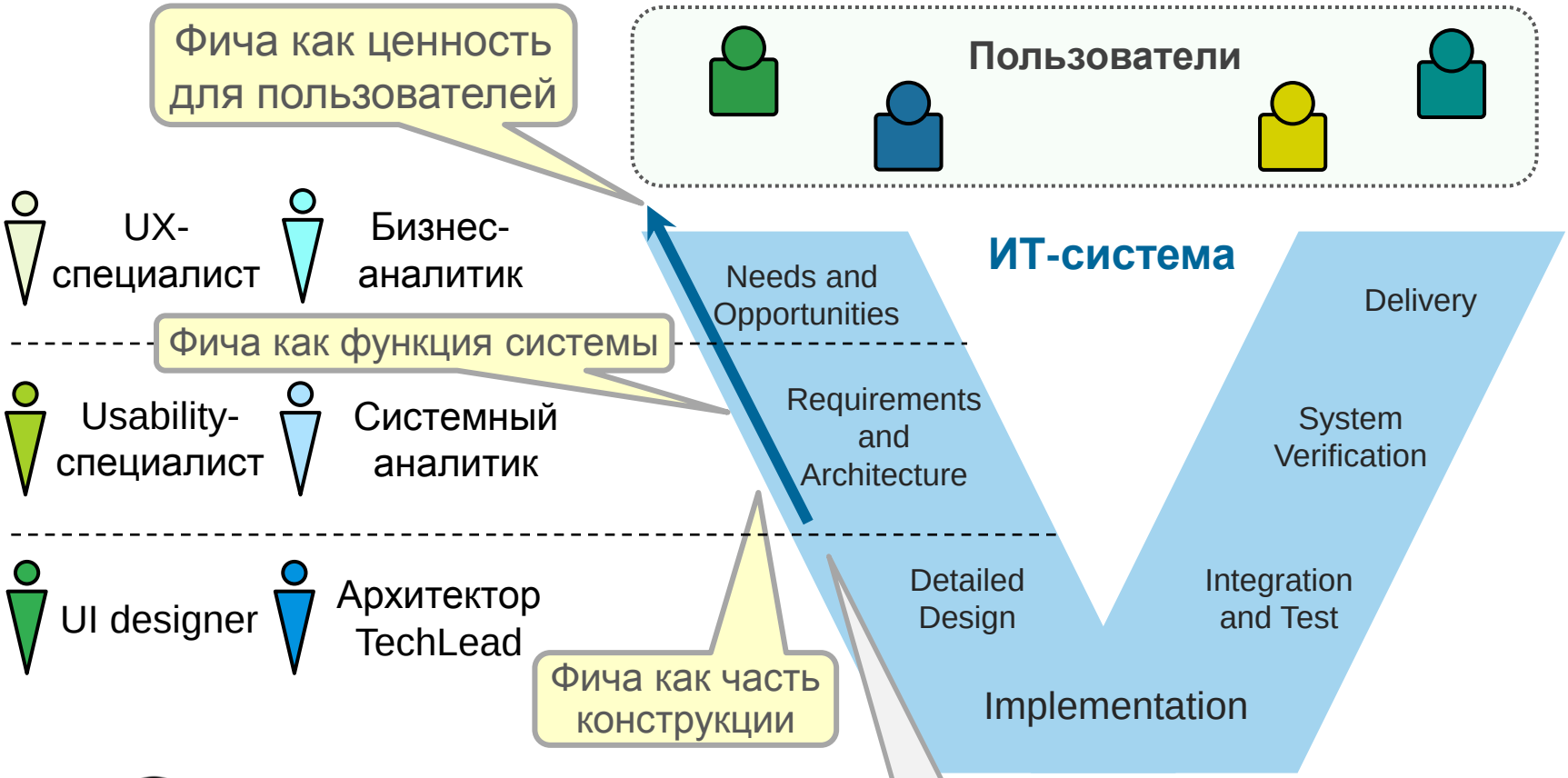
V-модель как карта проекта



Что является целью проекта?

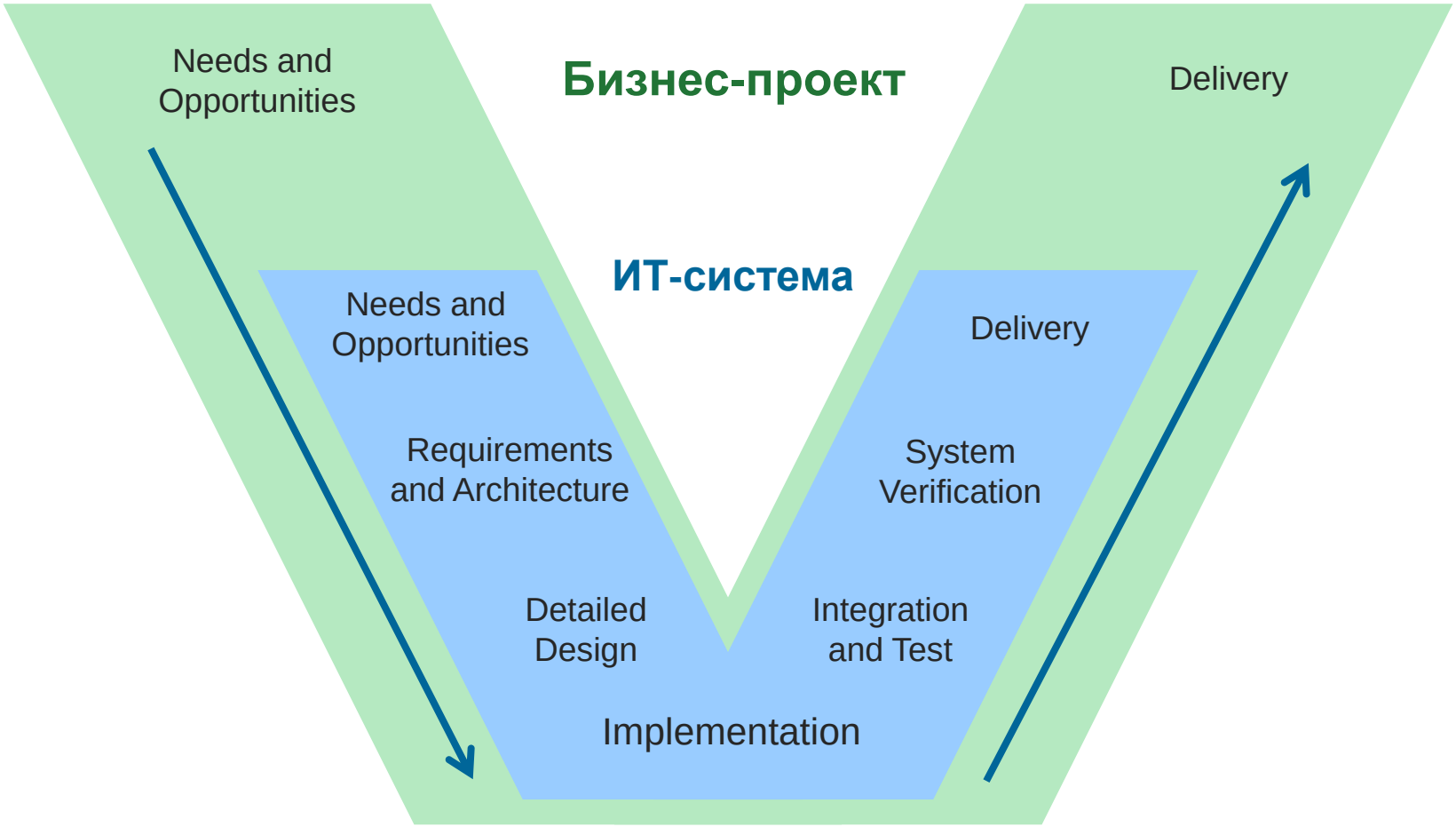


Уровни требований на V-модели



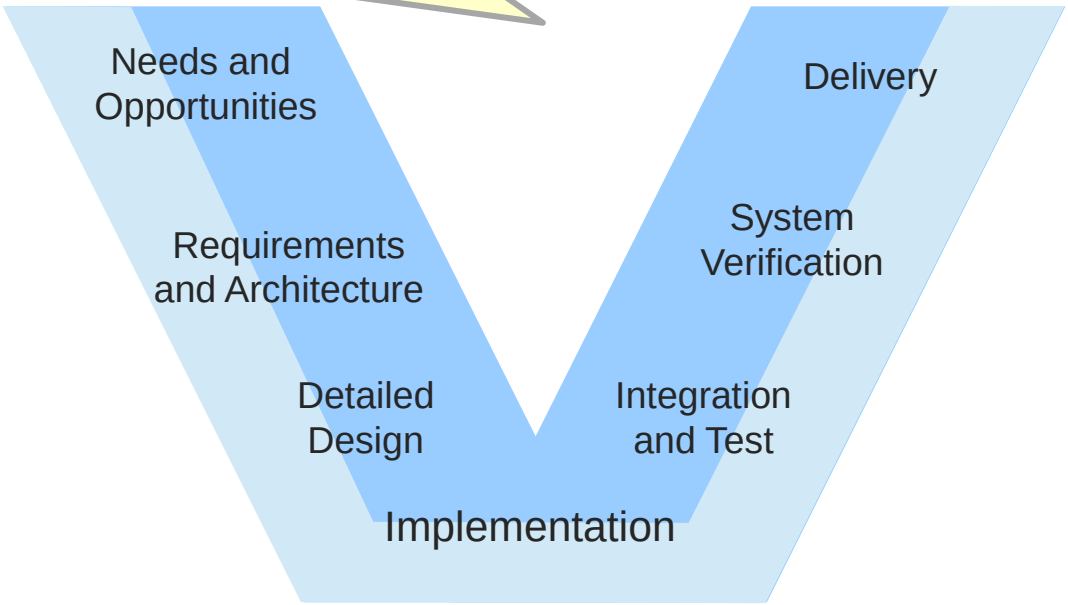
По мере развития IT **уровень требований** по V-диаграмме **повышался** и возникали **новые специализации аналитиков**

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!**



CUSTIS®

Требования и потребности – внешний контур проекта

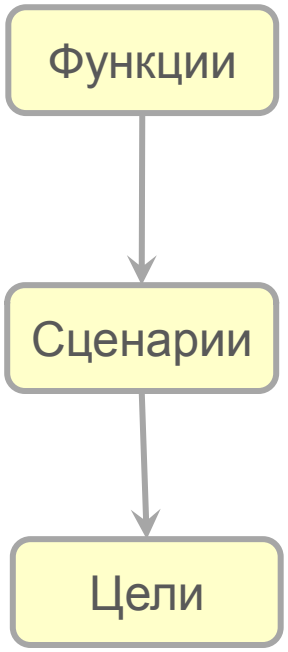
- Исходные данные для проекта – потребности пользователей или внешние функции?
- Насколько жестко надо фиксировать внешний контур проекта и трассировать к нему решения?
- Какие используем форматы: user story, use case, описания фич, классические требования?

Это зависит от стейкхолдеров, заказывающих проект, – как они видят границы и их описание?

CUSTIS®

Выйти за границу – представить себя на месте пользователя

- Первоначально требования описывали функции системы и систему как целое
- Выяснилось, что сделанные так системы оказываются неудобны или непригодны в работе
- Use case – описание, **как** пользователь будет применять систему для решения своих задач
- User story – фиксируем, **зачем** пользователь решает задачи, применяя систему



В процессе разработки принимается много решений, требующих **представить себя на месте пользователя**

CUSTIS®

Как декомпозируем
и инкрементально развиваем?

CUSTIS®

Компоненты, модули и их приращения

- Каждая система имеет деление
  - На **компоненты** – по внешним **функциям**
  - На **модули** – по **ячейкам** внутренней **конструкции**

  В общем случае они связаны произвольно

Из курса Левенчука, он ссылается на ISO 81346

- Развитие системы дискретно
  - Инкременты поставок очередных версий
  - Инкременты разработки, передаваемой в тестирование
- Если деление мелкое, надо удерживать целостность системы

Существует много вариантов организации системы из составных частей. Метод работы с требованиями должен соответствовать способу декомпозиции

CUSTIS®

Инкрементальная поставка

- Инкрементальная поставка требует создания бизнес-ценного функционала
- Формат **User story** формулирует **малый** функционал, ценный пользователю
- **Use case** больше, и его сценарии имеют разную ценность – делим case на slice

Ивар Якобсон
Use Case 2.0

- Можно описывать приращения в терминах доработки конструктивных элементов системы, ее фич, функций или сервисов

CUSTIS®

Инкрементальная детализация

- Детализация до мелких фрагментов выполняется не сразу – каковы крупные фрагменты?
- Соответствуют ли крупные фрагменты наборам поставляемого функционала или поставки формируются отдельно?
- На каких уровнях детализации и как оцениваем?
- Как организуем пакеты поставки из набора функционала?
  - Как выделяем MVP?
  - Как используем **приоритеты**?
  - Как удовлетворяем интересы групп пользователей?

Вариант – практика Story Mapping

CUSTIS®

Какова структура приложения?

- Что является поставляемым приложением?
  - Монолит, прирастающий по объему
  - Крупные модули
  - Мелкие сервисы или модули

Модуль – развиваем,
а **сервис** – заменяем новым

- Что является ячейками приложения?
  - Процедуры, таблицы и формы UI
  - Объекты (данные плюс методы в одном флаконе)
  - Модули с собственной моделью декомпозиции и др.
- Применяется ли слоевое деление и какое?
- Как интегрируются модули и мелкие ячейки?

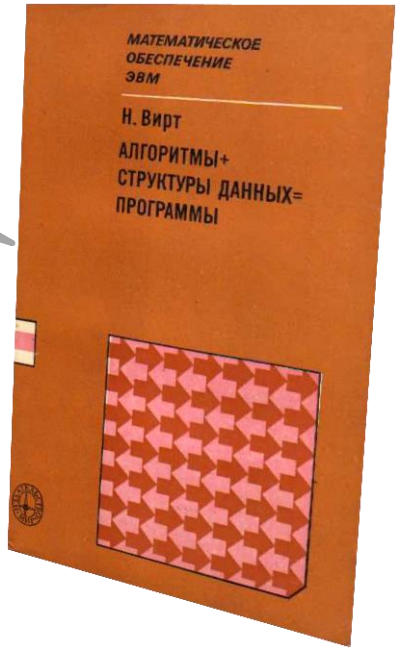
Основа структуры приложения

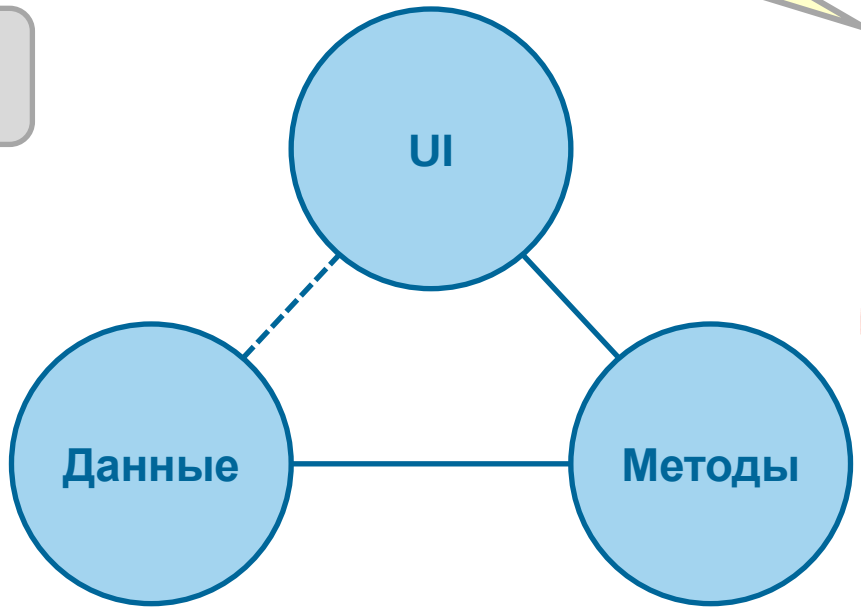
Интерфейс

Бизнес-логика

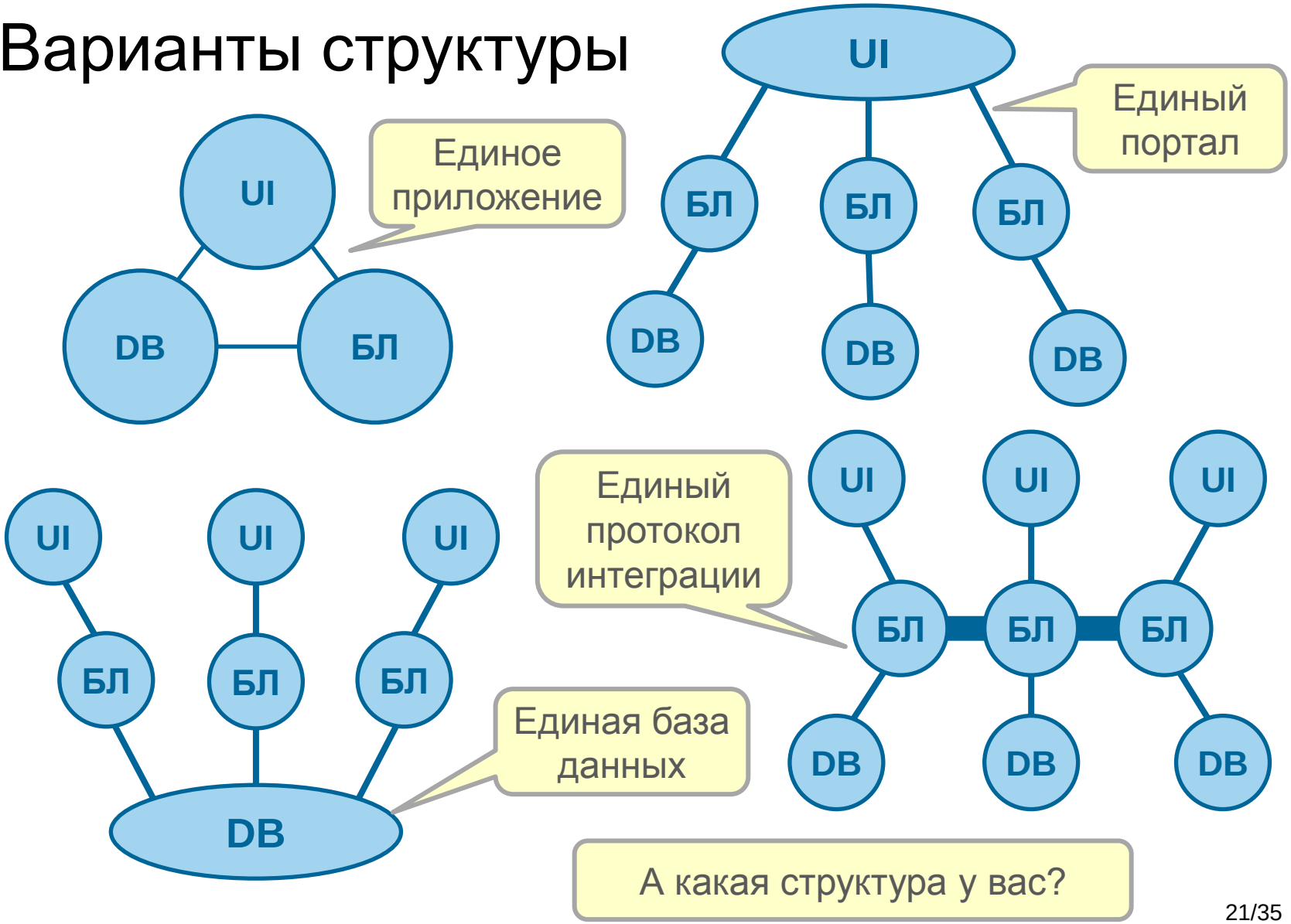
База данных

Николай Вирт
«Алгоритмы + структуры данных = программы»
1976 (на русском 1985)



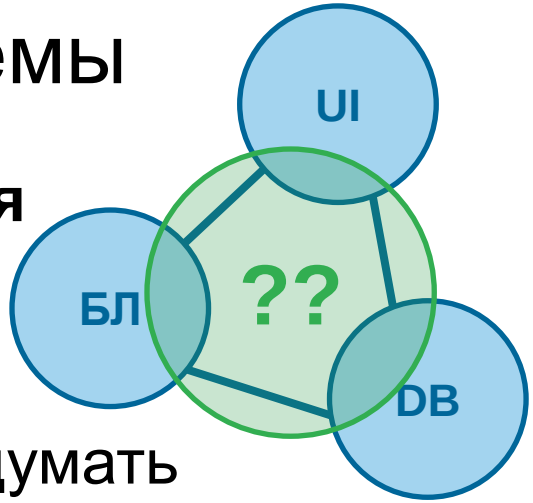


Варианты структуры



CUSTIS®

Удержание целостности системы



- **Концепция** системы – **цели создания** и верхнеуровневые полагания
- **Метафора** системы – эффективная практика XP, если ее получается придумать
  - Я говорил о метафоре в докладе «Модель системы – архитектура для Agile-разработки» на AgileDays – 2011
- **Архитектура** системы – основа конструкции
- **Модель** системы – постепенно создаваемая и принципиальная конструкция системы
  - Подход **Domain Driven Design (DDD)** адаптировал использование модели для инкрементальной поставки (мои доклады по DDD)

CUSTIS®

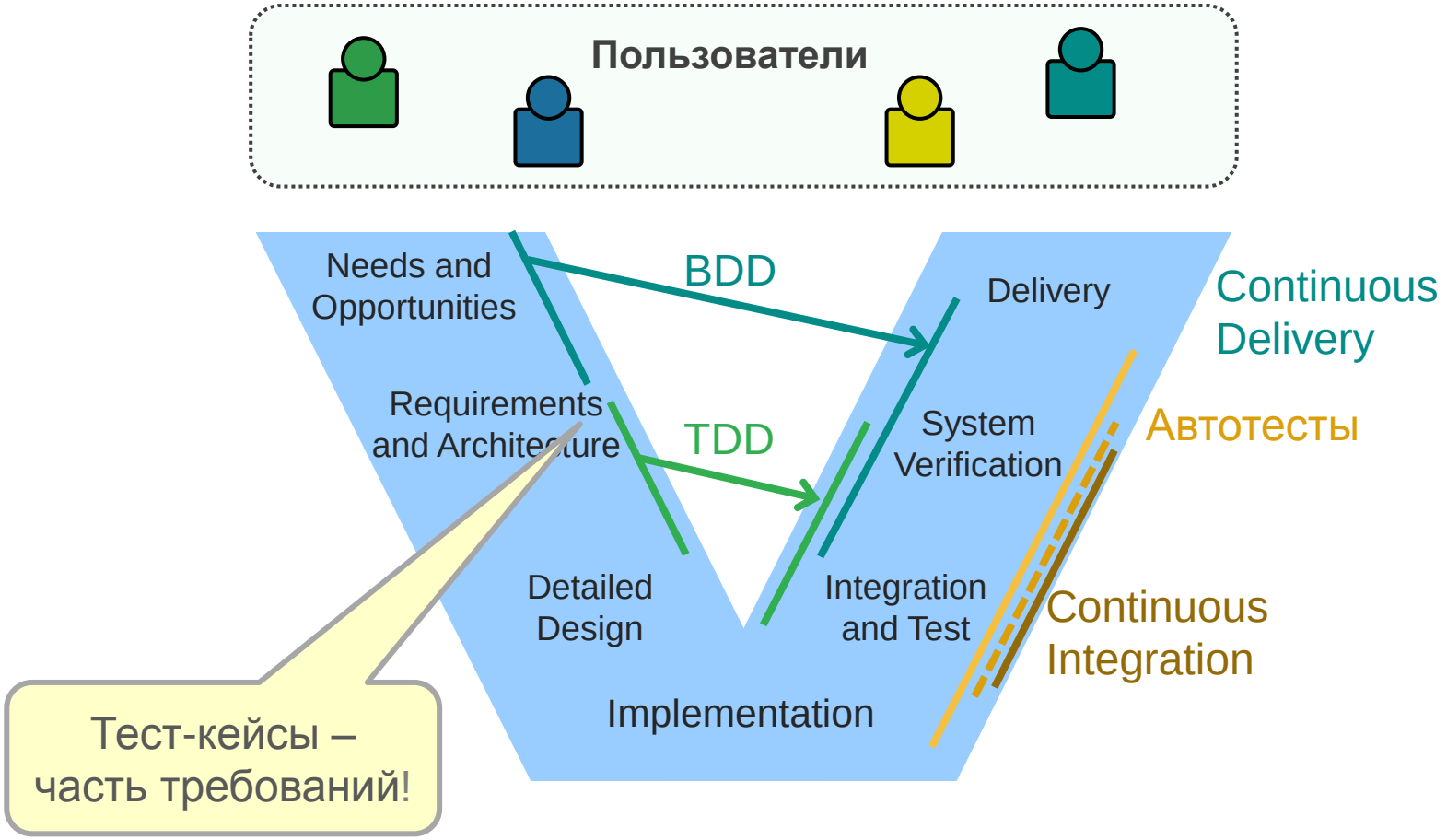
Как обеспечиваем качество?

CUSTIS®

Ограниченность изменений

- Инкремент поставки **развивает функции**, изменяя для этого ячейки конструкции
- Чтобы влияние изменений было локальным, ячейки должны быть изолированными
- ООП, микросервисы и многое другое придумывали для этого, но гарантий независимости нет
  - В ООП ее нарушают обращение по цепочкам ссылок, есть хороший доклад Андрея Бибичева на AgileDays – 2011 «Архитектура в Agile – переосмысляя идею модульности и компонентности»
  - Независимость микросервисов нарушает синхронное взаимодействие
- Какие практики будете использовать вы?

Практики тестирования



CUSTIS®

Ведение требований должно соответствовать подходу к тестированию

- Уровень тест-кейсов не может быть выше уровня требований – нет смысла делать BDD, если нужды пользователей неясны
- Уровень формальности требований должен соответствовать формальности тестов: автотесты требуют строгости
- Надо контролировать стоимость ведения и проверки тест-кейсов

CUSTIS®

Как организуем артефакты?

CUSTIS®

Артефакты и диаграммы

- Используем средство моделирования (EA и другие) или диаграммы «в вольном стиле»?
- Какие wiki-системы и другие средства коллективной работы используем?
- Как структурируем артефакты?
- Через какие viewpoint представляем систему?
- Какой набор диаграмм используем?

MS Word не является эффективным средством коллективной работы

На SECR – 2016 был рассказ Павла Музыки об опыте CUSTIS «Собираем кубик Рубика: восстановление архитектурного описания корпоративной распределенной системы»

CUSTIS®

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

CUSTIS®

От внедрения к эксплуатации

- Эксплуатация **добавляет** фокусы
  - Стоимости функционирования бизнес-процессов и поддержки
  - Быстрой и эффективной обработки инцидентов
- **Сохраняется** фокус развития системы
  - Автоматизация вспомогательных и побочных процессов
  - Реализация специальных процессов для конкретных целевых групп пользователей
  - Крупные доработки основного процесса
- Все это требует изменений в практиках и артефактах ведения требований

Их надо **спроектировать и реализовать**, тем более что на внедрении артефакты обычно отстают от системы

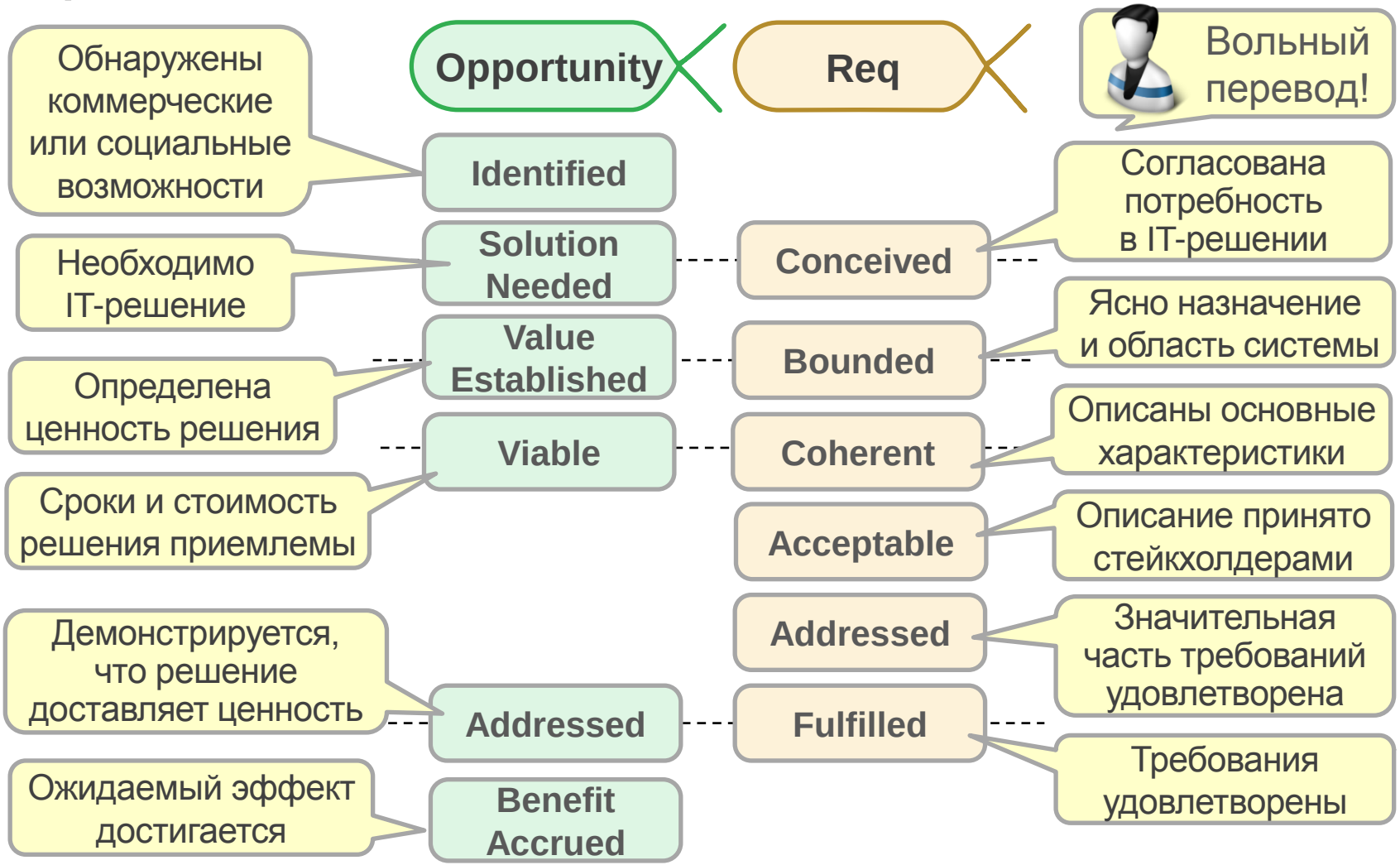
CUSTIS®

OMG Essence – способ описать метод

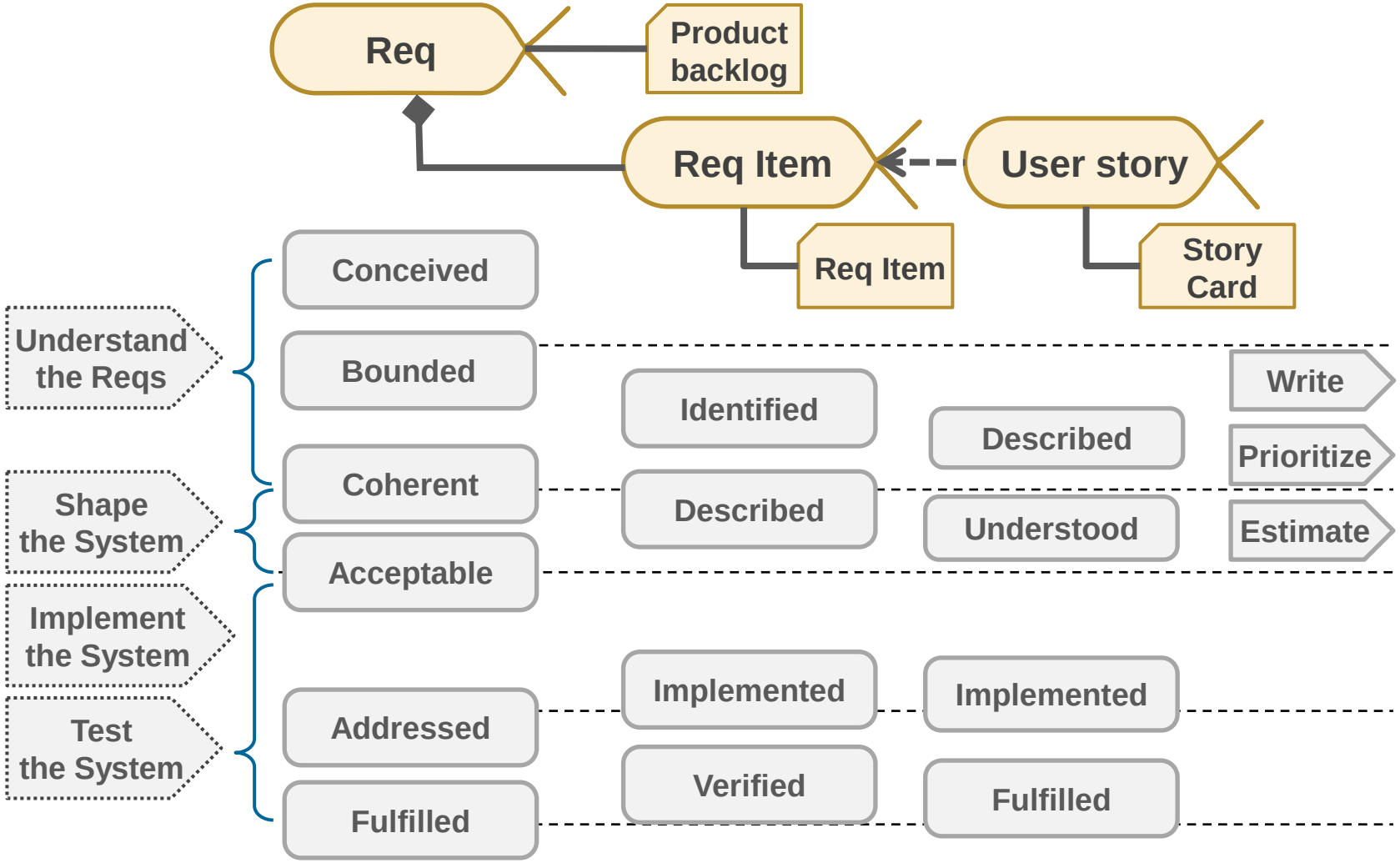
Потому что процессное описание в IT **не работает** – помни историю RUP

Источники: [Спецификация на сайте OMG](#) и [библиотека практик на сайте Ивара Якобсона](#) (требуется регистрация)

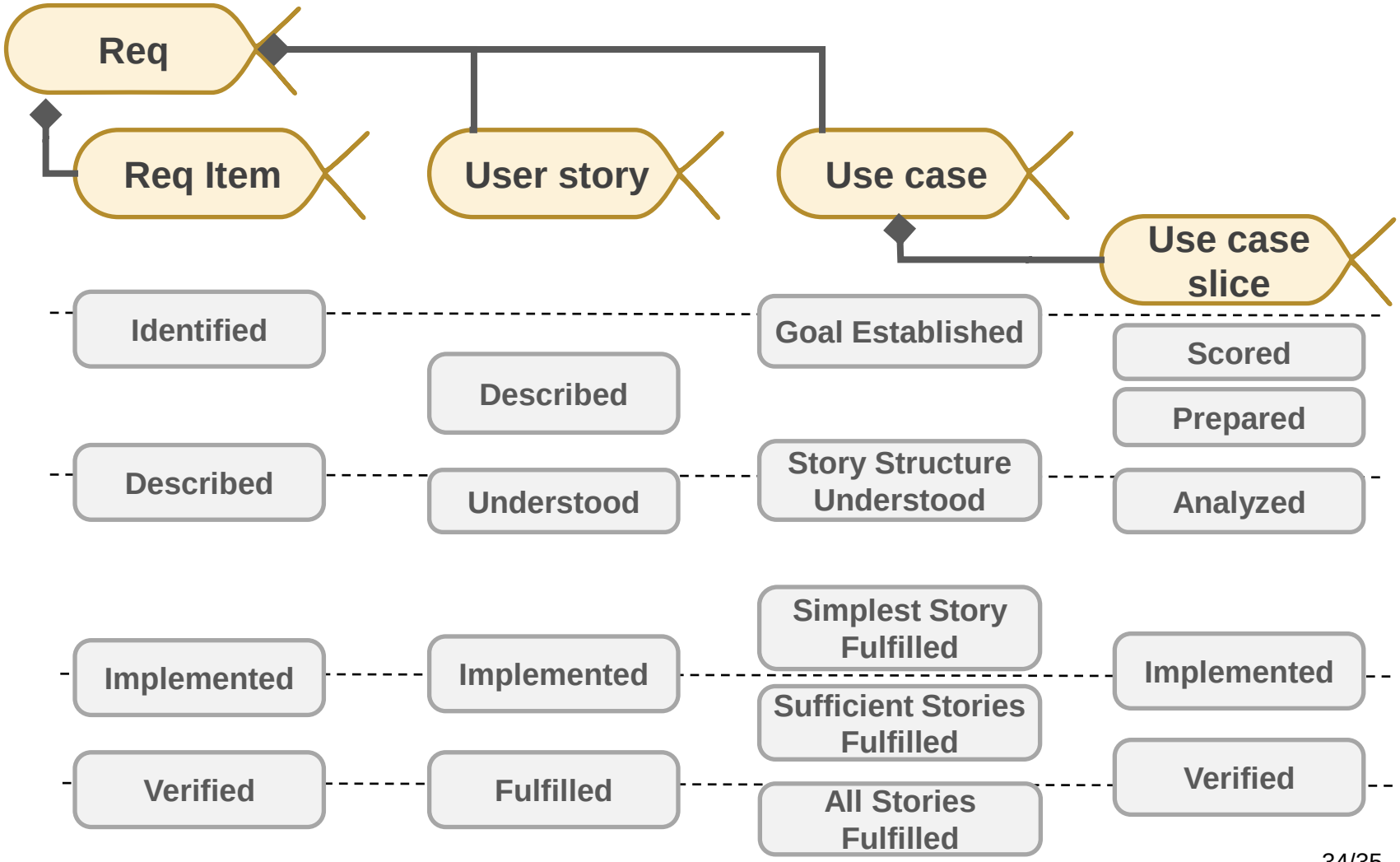
Требования и возможности в Essence



Требования – декомпозиция



User story и Use case



CUSTIS®

Подводя итоги

- Каждая ИТ-разработка идет в **своих условиях**
- Требования определяют внешние границы проекта и создаваемую конструкцию
- Способ работы с требованиями в проекте сам по себе – объект конструирования и воплощения

Максим Цепков
mtsepkov.org

Вакансии аналитиков

Пишите на hr@custis.ru, подходите с вопросами