

# Управление знаниями: какие документы нужны и что в них фиксировать



**Максим Цепков**

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

<http://mtsepkov.org>



## Teamlead Conf

Конференция для тех, кто хочет  
перейти на новый для себя уровень  
управления небольшими командами.



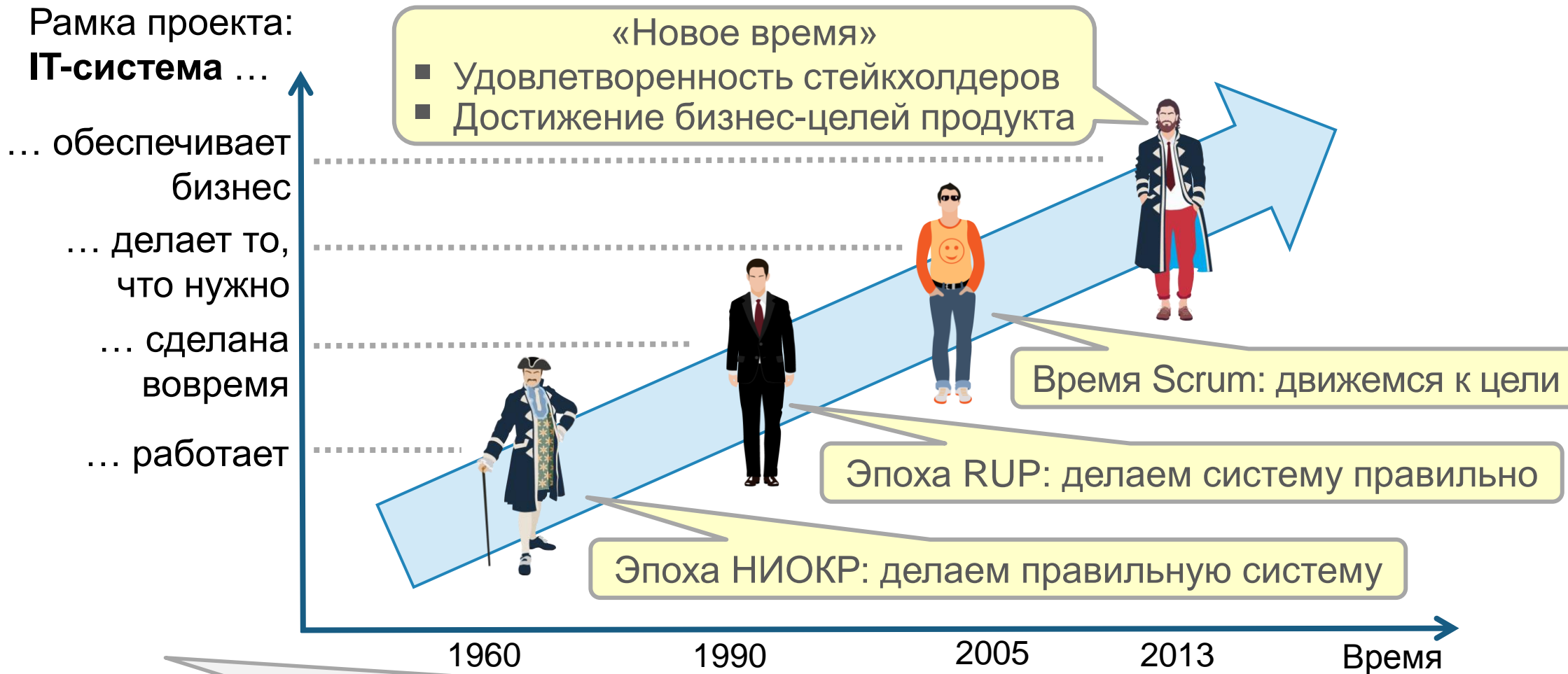
# Энтони Лаудер

## «Культуры программных проектов» (2008)

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

- История IT-отрасли делится на этапы
- Для каждого этапа характерен свой подход к ведению IT-проектов: свои представления об успехе, критерии качества, организации работ
- Выделил четыре культуры
  - Научная
  - Заводская
  - Дизайнерская
  - Сервисная
- Каждая культура породила **свои учебники**, они основаны на представлениях того времени и согласованы между собой

# Смена культур IT-проектов



Моя схема отличается от схемы Энтони Лаудера. Если схема Лаудера созвучна больше – используйте ее, только доведите до настоящего времени. Нет смысла выяснять, какая правильнее.

# Документы – какие они нужны?

- Основные учебники – эпохи RUP, ставка на универсальный рецепт
- Новейшее открытие: проекты – разные, и надо делать то, что уместно
- Не надо изобретать с нуля – используем шаблоны и образцы
- История аналогична интерфейсам: когда-то были представления об универсальном интерфейсе, сейчас – разнообразие, но есть style guide
- Разрабатывая style guide, ориентируются на UX пользователей, а на что ориентироваться, делая Doc Style Guide?



Я надеюсь, что доклад даст принципы и концепты, опираясь на которые, Вы сможете улучшать ситуацию и решать проблемы с документами в своем проекте

# Документы – для коммуникации

# Документ – средство коммуникации

- Документ не является самоценным, а обеспечивает коммуникацию
  - 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
- Форма документа выбирается, исходя из целей коммуникации

# Документ должен быть адресным

- **Не делаем** один документ для всех
- Делим документы по назначению и адресату
  - для принятия решений
  - текущей коммуникации
  - сохранения знаний во времени («мне через полгода»)
  - передачи знаний другим людям
  - помощи в текущей работе и др...
- Каждому назначению соответствует свой вид описания – viewpoint – и **свой метод описания**

# Критерий качества документа – уровень поддержки коммуникации

- Документ должен быть понятен всем сторонам коммуникации, и это ограничивает сложность нотаций
- Упрощенные схемы должны сохранять ключевые моменты
- Используем гипертекст и многообразие форм: схемы, текст, графики, аудио, видео



# Схемы и модели вместо текста

- Естественный язык многозначен и трактуем, а необходимо передавать смысл
- Применяем **схемы и визуализацию**, дополняя их текстовыми описаниями
- Описания не дублируют схему, а поясняют ее
- Создаем словарь понятий, **единый язык («ubiquitous language»)** проекта
- **Обсуждаем не термины, а содержание** – от тоталитаризма к плюрализму



UML прижился как схемы-картинки, а не как язык

# «Зачем» и «почему» важнее, чем «что»

- Классический use case отвечает только на вопрос **«что»**
- А в user story есть часть **«зачем»**:  
Как <роль> я хочу <сделать что-то>, для того чтобы <достичь целей>
- Она появилась не сразу, а из опыта работы
- И позволяет разработчику поставить себя на место пользователя, если надо выбирать варианты решения



Этот урок надо учитывать, даже если используешь другие форматы – фиксировать цели пользователей и бизнеса, начиная от целей проекта. А то содержание целей проекта часто сводится к «бизнесу зачем-то надо...»

Проектируем документы проекта

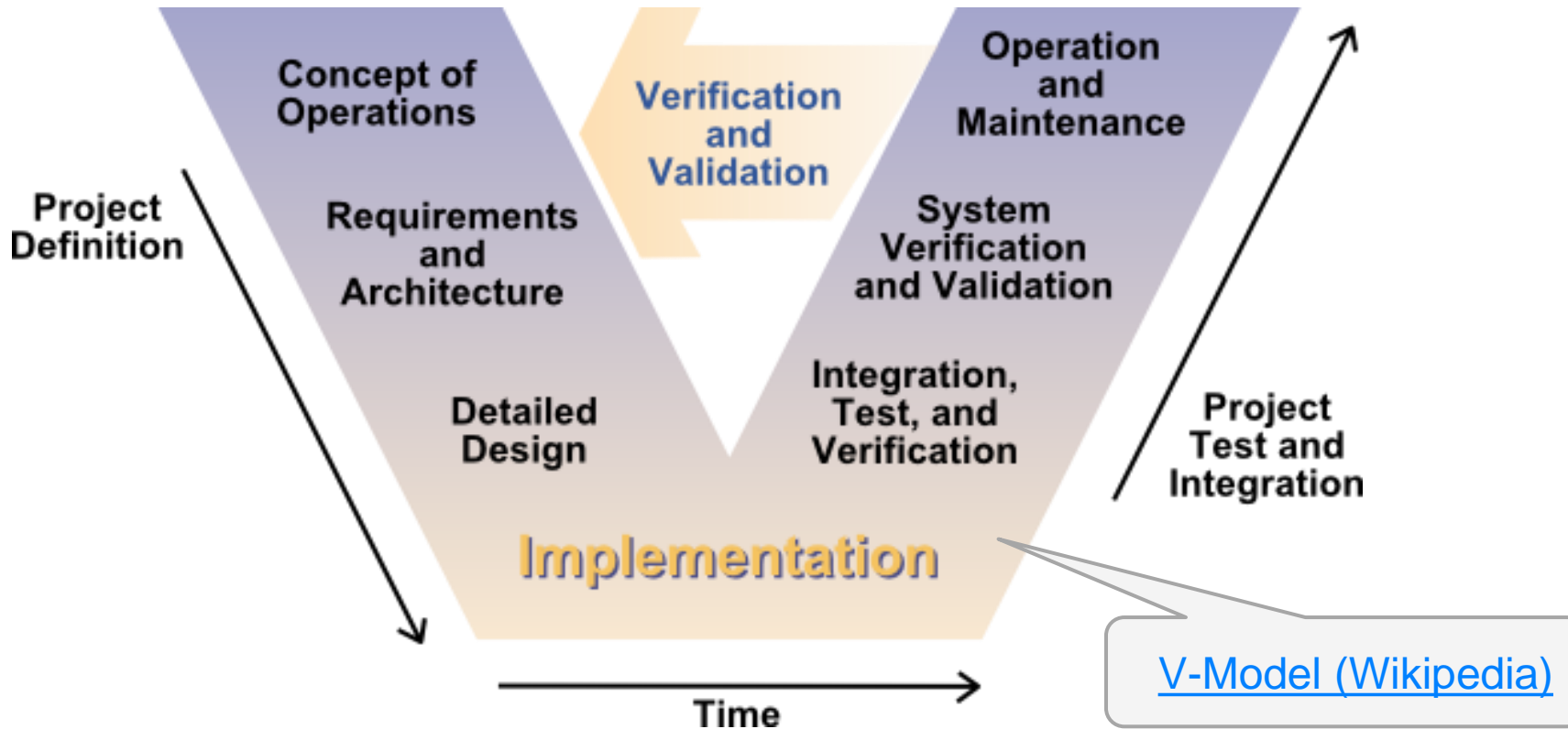
# Как проектировать? – Как любую систему!

- Выделяем кейсы коммуникаций
- Определяем документы, которые их поддерживают
- Если коммуникация распределена во времени – то нужны ответственные за поддержание документов
- В процессе использования – оцениваем качество и улучшаем

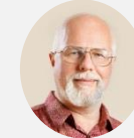


Похоже на проектирование интерфейсов от Usability-UX

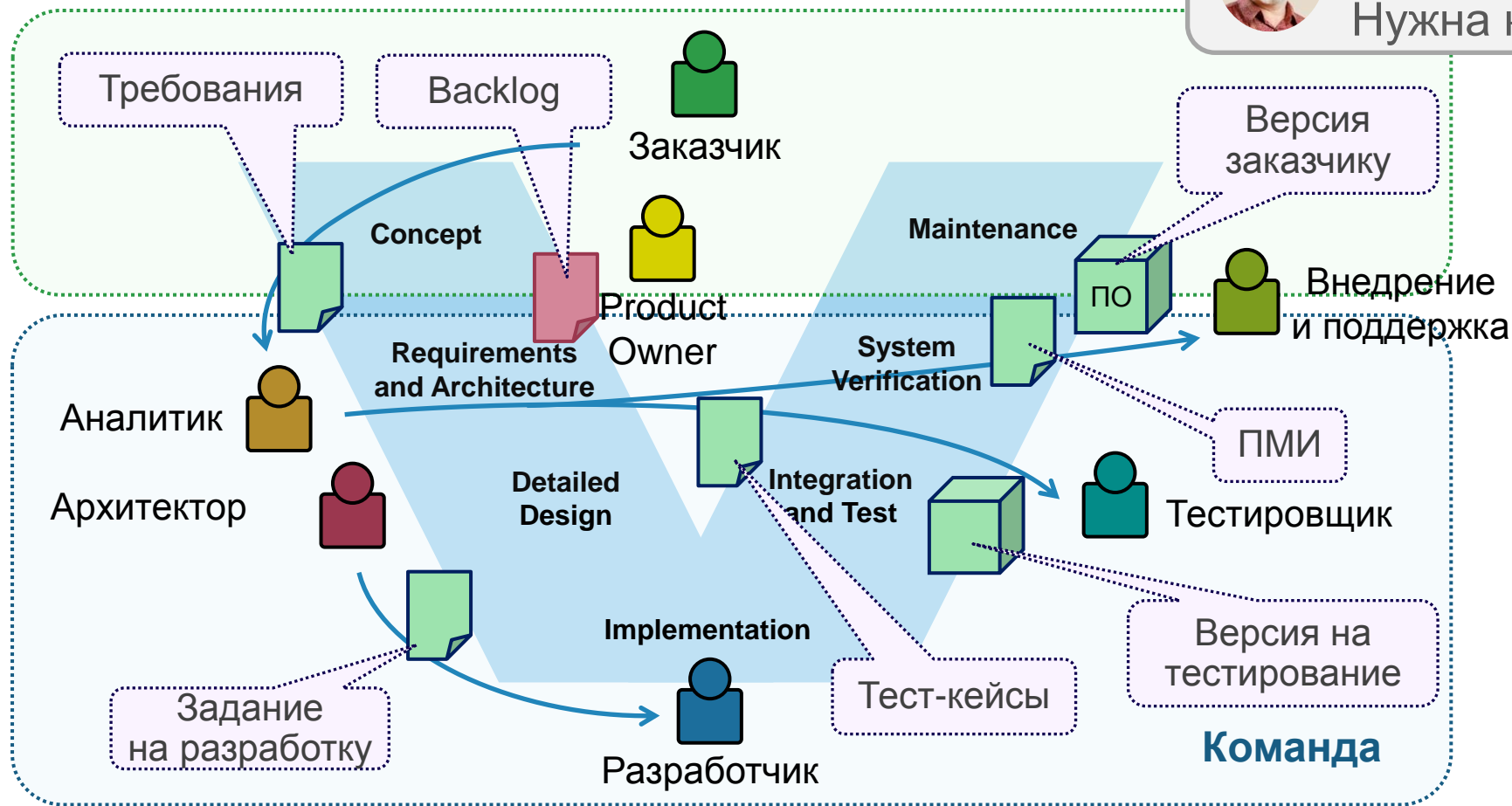
# Представление о проекте – V-модель



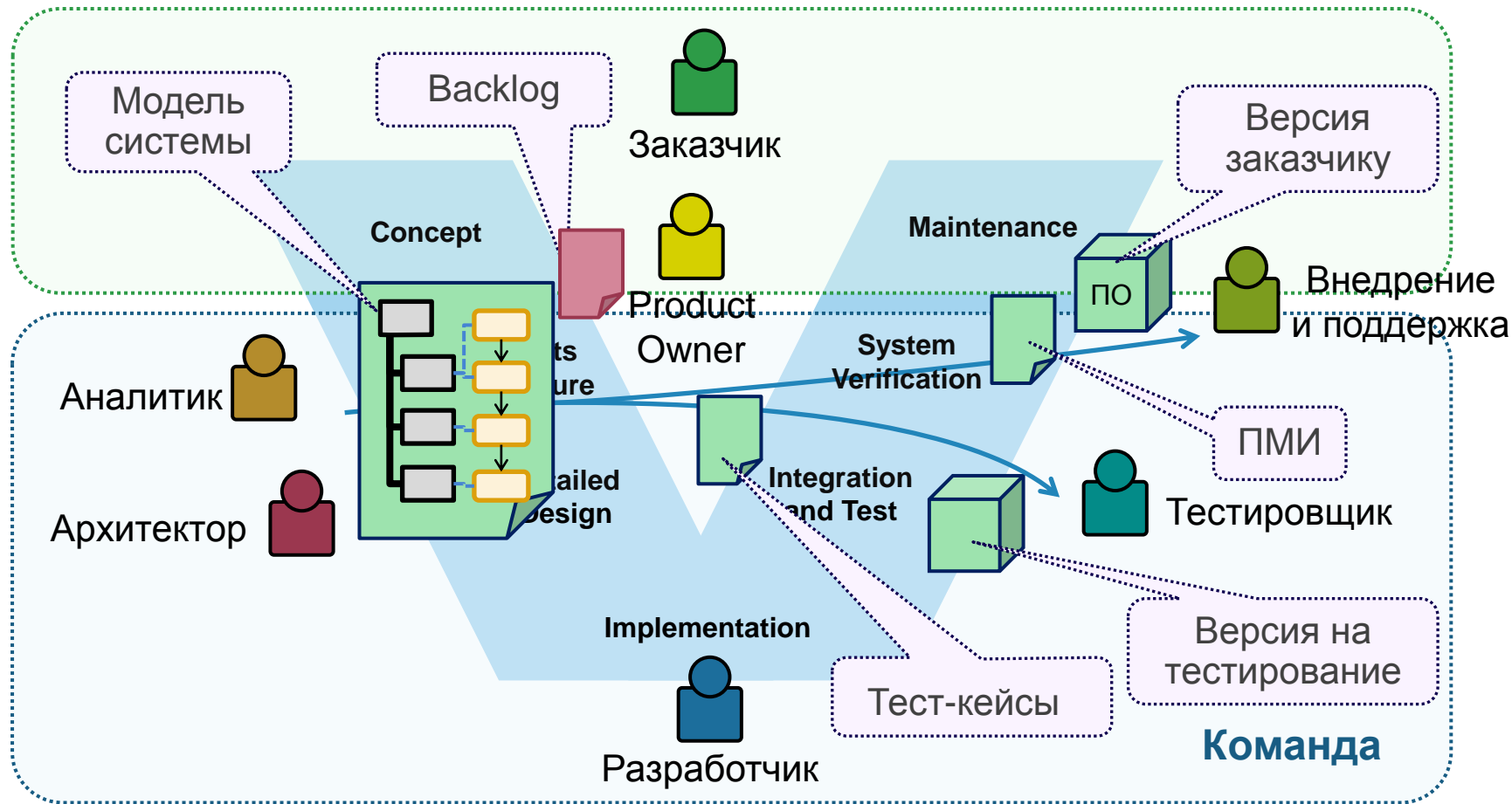
# Классические артефакты – на границах ответственности



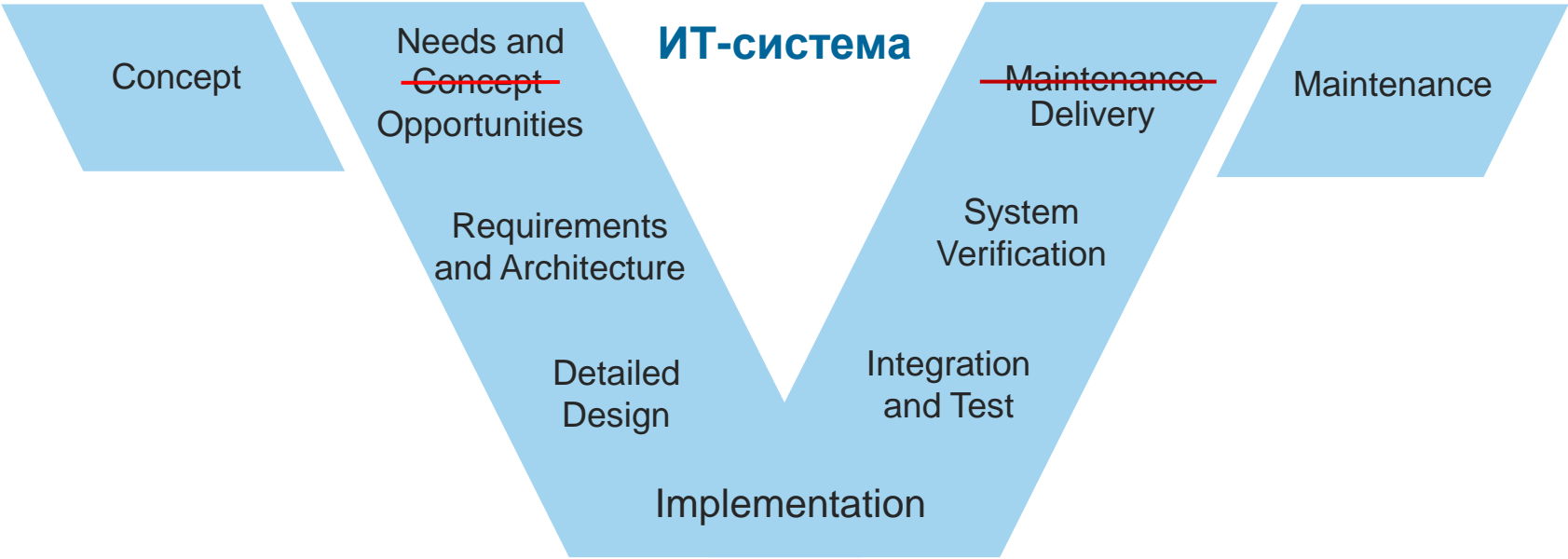
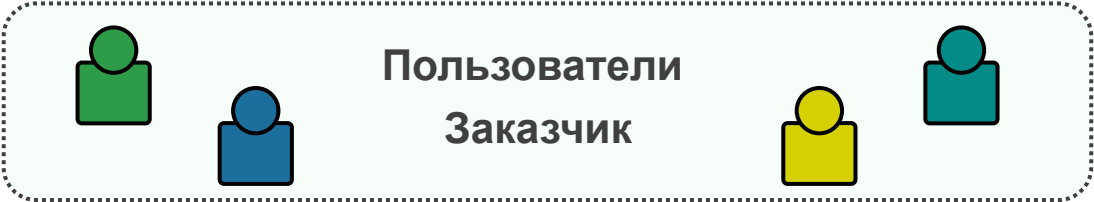
Опыт IT: передача через артефакты – не работает. Нужна коммуникация вокруг



# DDD: модель – общий разделяемый артефакт



# Современный проект – расширенная V-модель





# Артефакт *концепция* или *vision*

Фиксирует стартовые полагания проекта

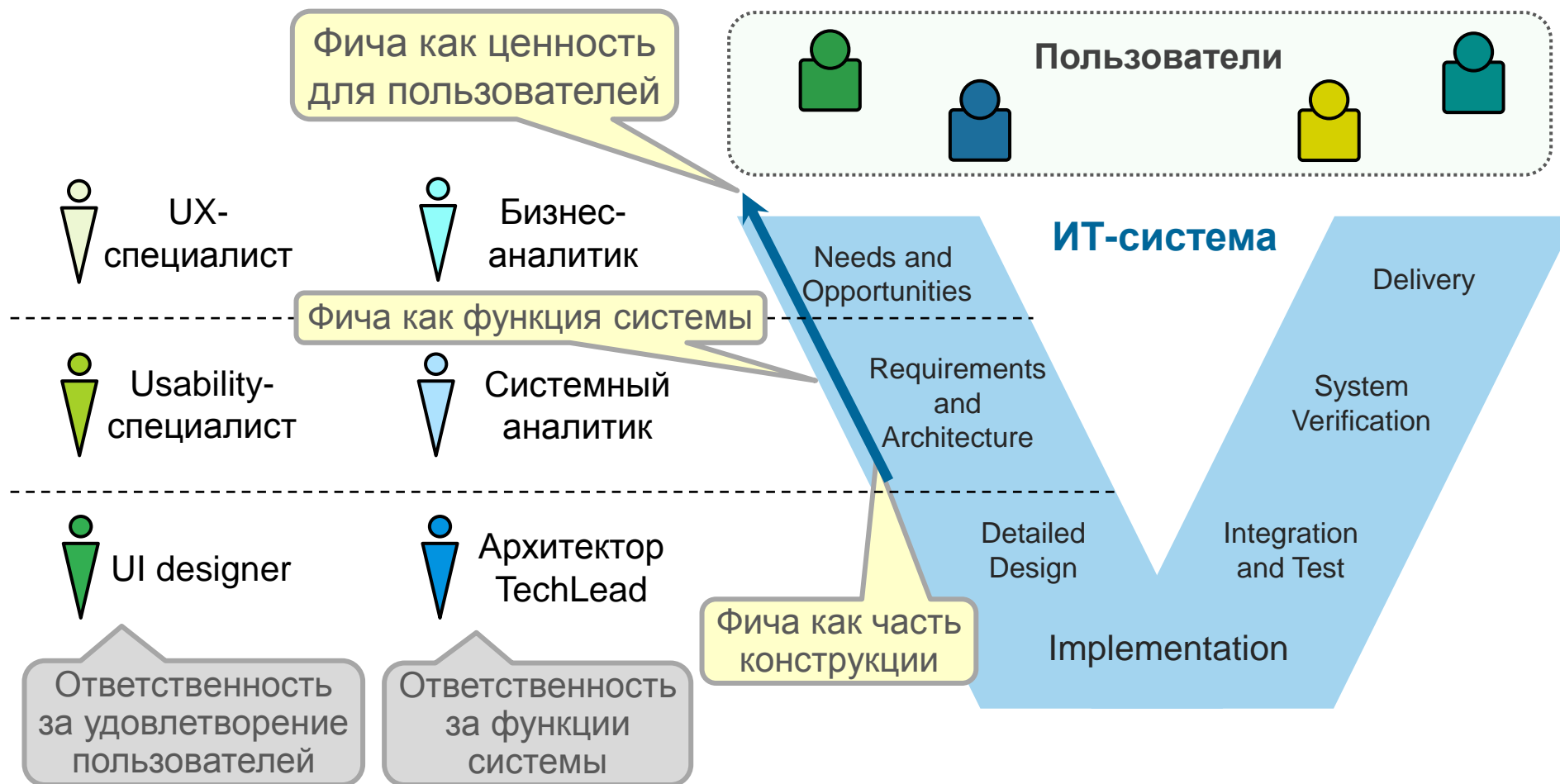
- Идея проекта – **возможность** для бизнеса: что, для кого и как сделать
- Реализуемость и целесообразность идеи
- Интересы и ожидания основных стейкхолдеров от результата проекта
- Устройство продукта и его использование
- Оценка необходимых для проекта ресурсов

С концепцией соотносимся по мере развития проекта



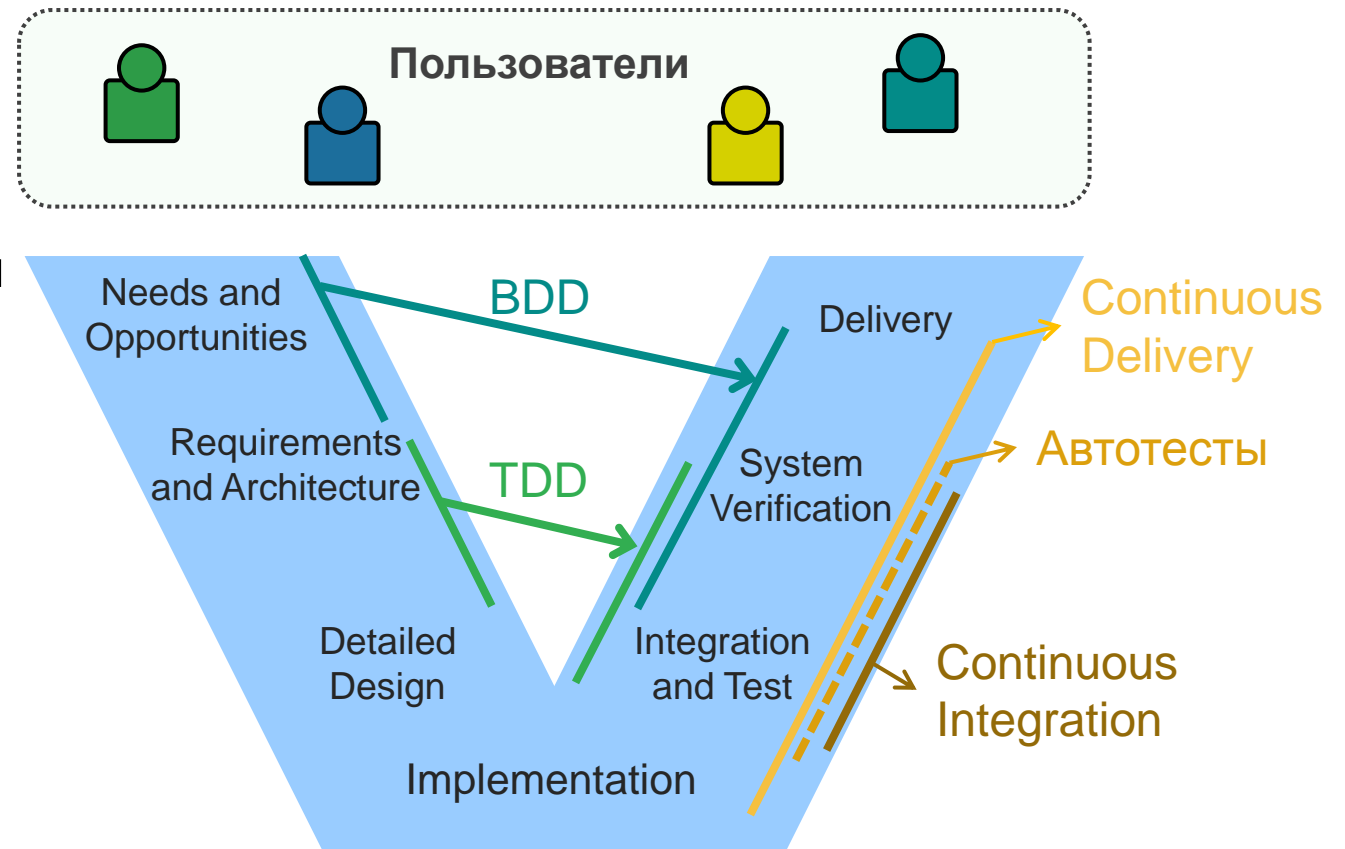
Необходимая **подробность** определяется функционально:  
**что стейкхолдерам нужно знать** для решения о старте проекта

# Внешняя граница проекта



# 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



# Артефакты для индивидуального фокуса

- Часть артефактов всегда перед глазами – для постоянного индивидуального фокуса члена команды
  - Отражение движения проекта: Burn down chart, Доска с задачами
  - Основные концептуальные схемы – на стенах
  - Цели и идеи, которые воплощаем сейчас
- Материальное представление обычно эффективнее электронного для команды, но недоступно удаленно и его нельзя агрегировать

# Точки коммуникации для формирования знаний о движении проекта

- Демо – представление состояния проекта тем, кто пользуется результатами твоей работы, **и другим интересующимся**
- Ретро – оценка себя: правильно ли мы работаем и что можно улучшить
- Daily meeting – синхронизация представлений команды о движении проекта
- Планирование – синхронизация намерений

У каждой встречи – свое назначение и соответствующий ему формат

# Долговременные описания

- Задача описания – передать знания в будущее
- Какие кейсы поддерживает описание?
  - Научить нового пользователя работать с системой
  - Дать справочную информацию по редким функциям системы
  - Представить концептуальное устройство системы новому разработчику
  - Помочь вспомнить устройство фрагмента системы автору для доработок
  - Описать устройство модуля системы для развития новым разработчиком
  - Дать информацию об устройстве системы инженеру сопровождения
  - ...
- Универсальное описание – детальное, дорогое и не актуальное
- Надо фиксировать назначение и оценивать по соответствию ему

# Документ имеет цену

- Чем подробнее документ, тем он дороже
  - Насколько подробны должны быть описания устройства бизнеса?
  - Протоколы совещаний – понятны присутствовавшим или отсутствовавшим тоже?
- Поддержание документа актуальным тоже имеет цену
- Управляем детальностью и актуальностью документов
- Используем компромиссные варианты: резюме + видео/аудио
- Не забываем фиксировать основания и логику решений – их упускают чаще всего

# Нормативно необходимые документы

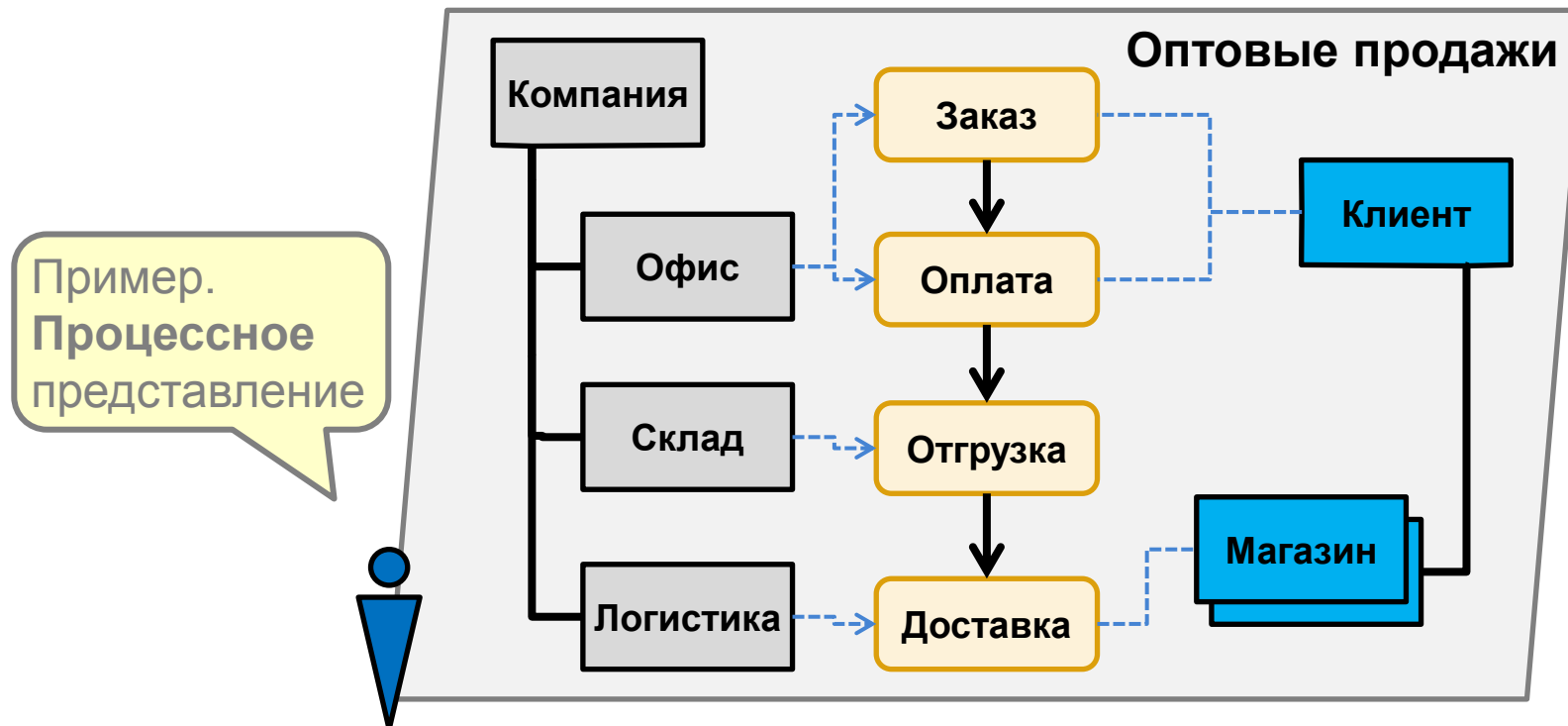
- Обеспечивают коммуникацию Заказчика с его проверяющими
  - Выясняем у Заказчика требования на документы для успеха проверки
  - Делаем check list – и выполняем
- Использует ли их Заказчик для других целей?
  - Нет, только для проверяющих (write only docs) – делаем отдельный комплект, соответствующий требованиям
  - Да, сам Заказчик по ним работает, считая это разумным – применяем check list к рабочим документам проекта
  - Промежуточный вариант: одни – только для проверок, другие – рабочие
- Создаем документы, адекватные цели использования



Коммуникация передает смыслы

# Разложить по полочкам

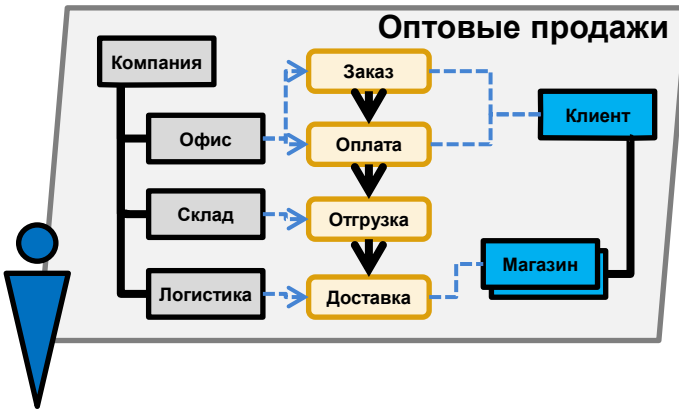
- Работа аналитика начинается с согласования словаря понятий предметной области и ее модели
- Предполагается, что этого **достаточно** для коммуникации с заказчиком на общем языке



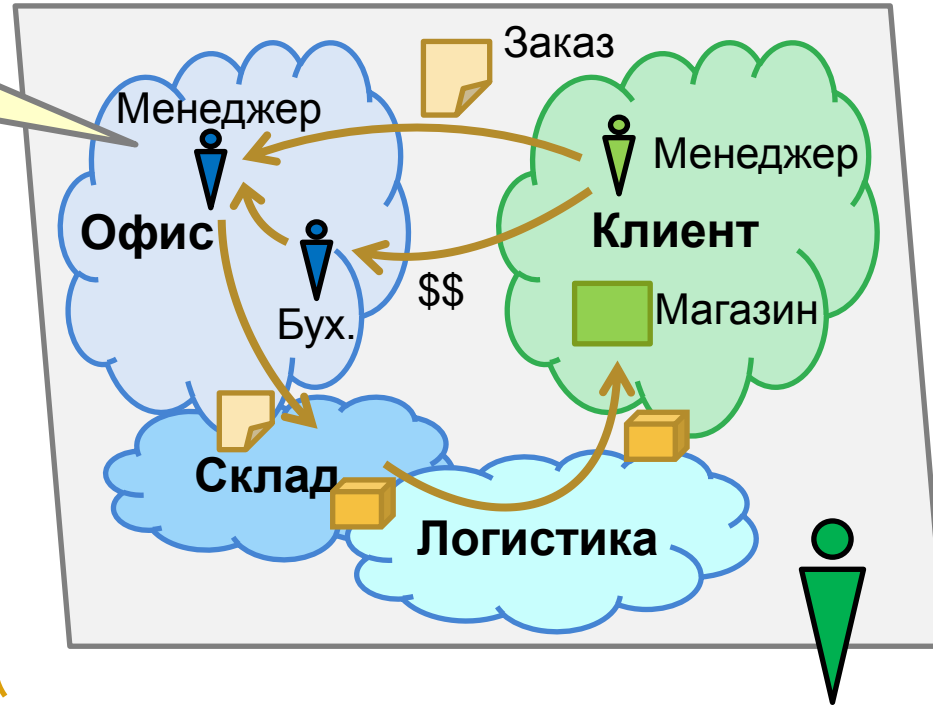
# А у заказчика – все по другому

Модель взаимодействующих агентов. Важно совсем не то, что в процессной модели.

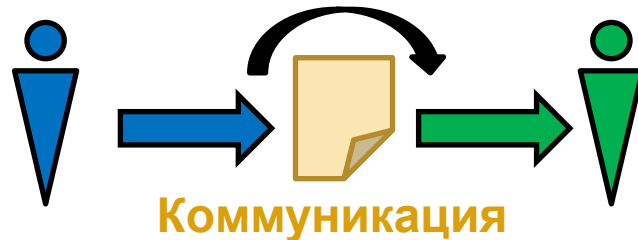
## Я и мое мышление



## Партнер и его мышление

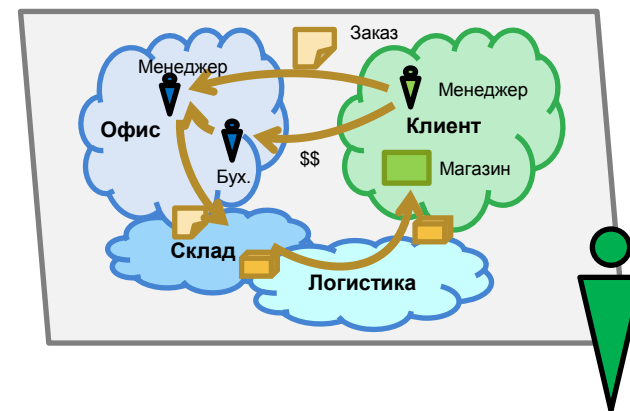
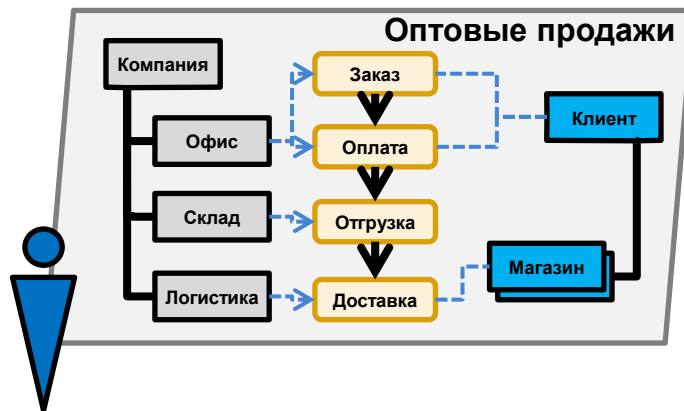


понимание?

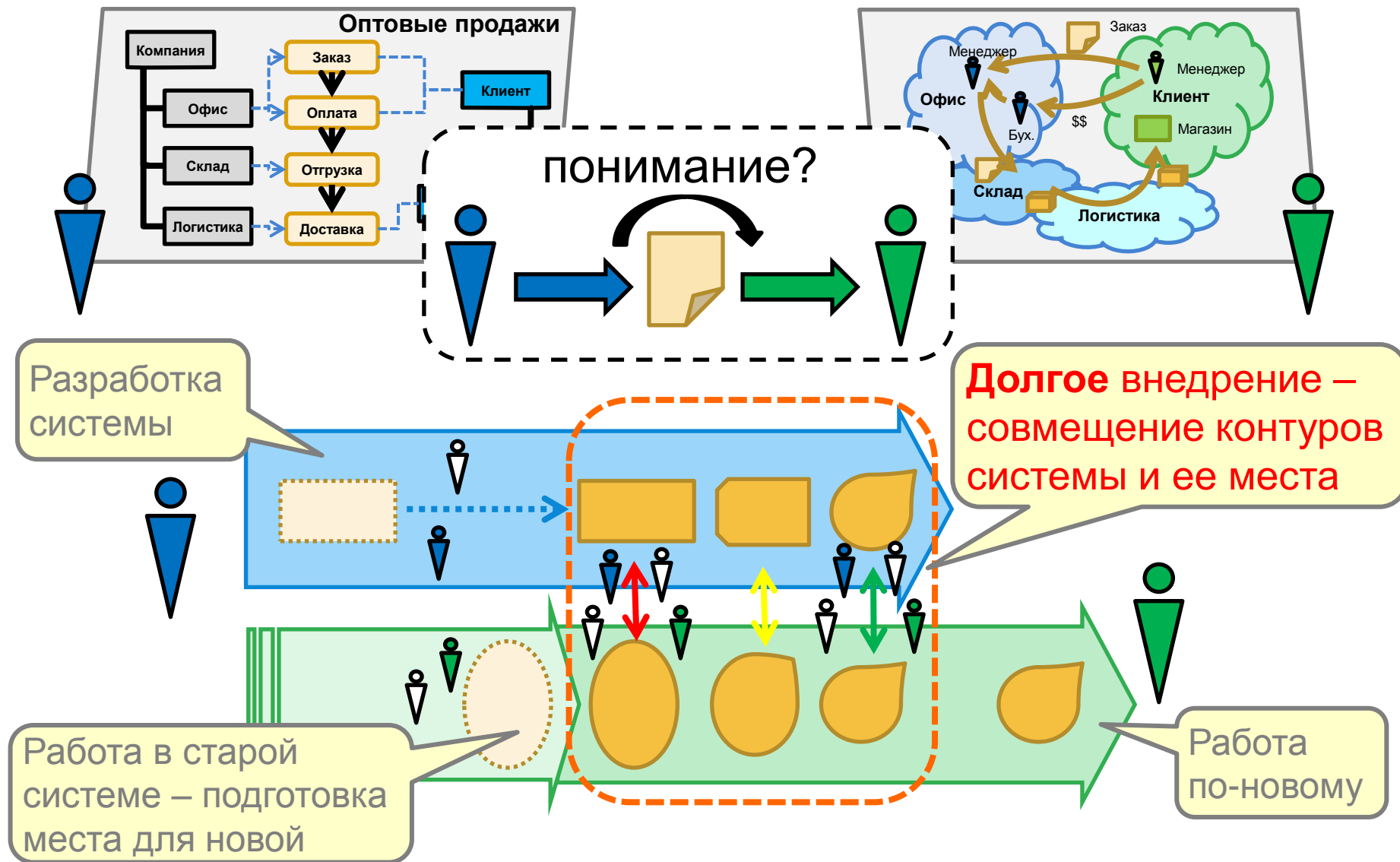


# Разный способ мышления – норма

- Иерархические модели, **таксономии** – результат **научного освоения мира**, в котором все приведено к строгой форме и «разложено по полочкам»
- Ребенок не может «разложить по полочкам», мыслит **облаком тэгов**, понятий, связанных ассоциативными цепочками – **фолксономиями**
- Такое мышление эффективно и операбельно, на нем построены маркетинг, масс-медиа и политтехнологии



# Если разницу не поняли сразу...



# Что делать? – Рисовать схемы!

- **Схема** задает форму представления информации
- Рассказывая схему, ты передаешь свои представления об объектах окружающего мира
- Представьте предыдущую часть без схем, текстом...
- В IT сила схем давно известна, но восприятие надо проверять
  - Но формальные нотации хорошо понимаются далеко не всеми
  - А эскизные, неформальные схемы многие воспринимают как картинки
- За пределами IT есть источник схем – СМД-методология, смотри мой доклад «[Коммуникация при различной структуре мышления - таксономия против фолксономии \(AnalystDays-2016\)](#)»

# Принципы ведения документов

# У документа нет автора

- Документ живет дольше его первого автора
- **Работаем коллективно, а не пересылаем**
- Используем **wiki-системы** – они позволяют строить системы связанных документов
- Google Docs и аналоги тоже можно использовать, но они хуже, т. к. ведут отдельные документы
- Увидел, что улучшить, – сразу сделал, **согласование** – только **по несогласию**
- Правим ответственно и уведомляем



# Документ создаем постепенно

- Большой документ устаревает раньше, чем будет написан
- Концепты – кратки, проводим детализацию по необходимости, а не сразу
- Делаем ту часть документа, которая касается текущей задачи
- Используем форматы, ориентированные на инкрементальное создание документов и софта – user story, slice use case, story mapping

# Содержание важнее формы

- **Не работают** формальные требования к документу – оглавления, обязательная форма заполнения
- **Работают** критерии пригодности документа к использованию стейкхолдерами
- Готовность оцениваем экспертно
- В помощь экспертам – check list проверки

# Оставляем следы

- Протоколы совещаний, резюме разговоров **необходимо** оформлять документами
- Задачами управляем не в переписке, а в системах ведения дел
- Материалы доступны всем участникам работы, есть поиск и навигация
- Цель следов – это не поиск виноватых, а восстановление обстоятельств и действий

# Типовые модели знаний

- Модели для описания бизнеса в Archimate
- Модель мотивации стейкхолдеров в Archimate
- Подходы объектно-ориентированного программирования, перенесенные на разработку онтологий в Domain Driven Design
- Карта ведения проекта OMG Essence
- Схема множественных viewpoint'ов ISO 42010



Они слишком тяжелы, если **соблюдать форму**, но хороши для проверки содержания и структуры

# Характер работы определяет инструменты

- Характер работы с документами определяет инструменты документирования, а не наоборот
  - Так же, как характер интерфейсов определяет фреймворк их разработки
- Хорошие варианты
  - Все в вики: общее пространство документов, коллективная работа, ссылки, легкость изменения, поиск, схемы и медиа (например, <http://4intra.net>)
  - Кратковременные документы – рядом с задачами (можно в Google Docs), а долговременные – в публикуемой документация на систему вместе с кодом
- Главное – не система документирования, а способ ее использования

Управляем профессиональными  
знаниями

# Используем все каналы

- Знания поступают по многим каналам
  - Тематические интернет-порталы
  - Социальные сети и группы в них
  - Online- и offline- конференции и семинары
  - Meetup'ы и встречи профессионалов – быстрые знания
  - **«Встречи на кухне» на работе**
- Выбираем эффективные для себя каналы
- Комбинируем разные формы получения знаний
- Ведем активные коммуникации:  
**чтобы получать знание, надо его отдавать**

# Слушаем пульс времени

- Знания о технологиях и способах работы меняются очень быстро
- Практика сильно опережает теорию
- Знания не успевают оформляться в «солидные и проверенные» источники
- Надо слушать пульс времени, быть в курсе нового, пробовать применять его



Казалось бы, очевидно.  
Но многие по-прежнему ждут, когда выйдет учебник...



# Сообщества – в компании и за пределами

- Встречи на кухне – хорошо, но недостаточно, знания не сохраняются
- Общая wiki – тоже недостаточно, знание без коммуникации мертво
- Надо организовывать обмен профессиональными знаниями в компании
- Не факт, что внутри достаточно компетенций – часть знаний получается через включение во внешние сообщества
- Работа сообщества и накопление знаний имеет цену – надо выделять ключевые и управлять процессом

# Уроки управления знаниями

- Знания могут быть – но как найти человека?

Урок от BBC. При командировке за новостями важны знания о местном контексте, которые есть у любого, кто был. Поэтому важнее не профессиональные сообщества, а общая база сотрудников с информацией о поездках

- Профессионал не понимает вопроса дилетанта – нужны переводчики

Урок от IBM. Возможность просто задать вопрос профессионалу – недостаточно. Профессионал часто просто не опознает вопрос. Отдельная компетенция – понять по вопросу тему, а потом перевести вопрос эксперту по теме, чтобы тот принял его

- Управление знаниями – отдельная отрасль со своими практиками

- Крис Коллисон, Джефф Парселл «Учитесь летать»
- Мои отчеты по конференциям Knowledge Management Russia  
<http://mtsepkov.org/KM>

# Подводим итоги

- Нет идеальной системы документов, к которой можно стремиться
- Систему работы с документами строим для проекта
- В докладе были принципы, которые могут быть ориентиром в создании конструкции системы документов
- Оценивайте, насколько система документов адекватна проекту
- В существующем проекте лучше идти от болей и проблем, улучшая
- В новом проекте – можно спроектировать заново



Максим Цепков

<http://mtsepkov.org>  
[maks.tsepkov@ya.ru](mailto:maks.tsepkov@ya.ru)

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