

# UX: делаем – систему удобной? Нет. Делаем удобной жизнь!



Максим Цепков

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

<http://mtsepkov.org>

World Information Architecture Day  
Санкт-Петербург 24.02.2018

# Исторический экскурс

# 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

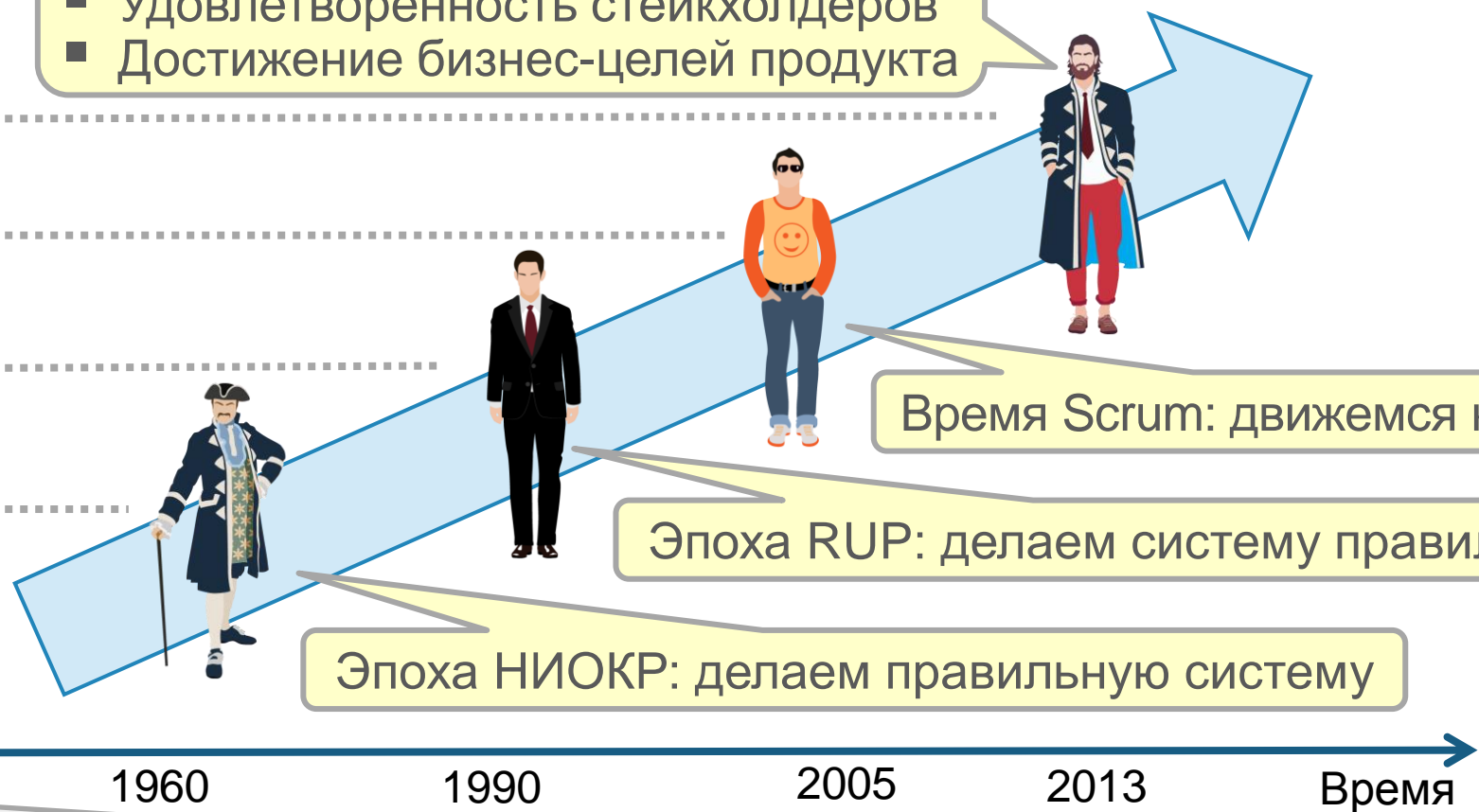
# Смена культур IT-проектов

Рамка проекта:  
IT-система ...

... обеспечивает бизнес  
... делает то, что нужно  
... сделана вовремя  
... работает

«Новое время»

- Удовлетворенность стейкхолдеров
- Достижение бизнес-целей продукта



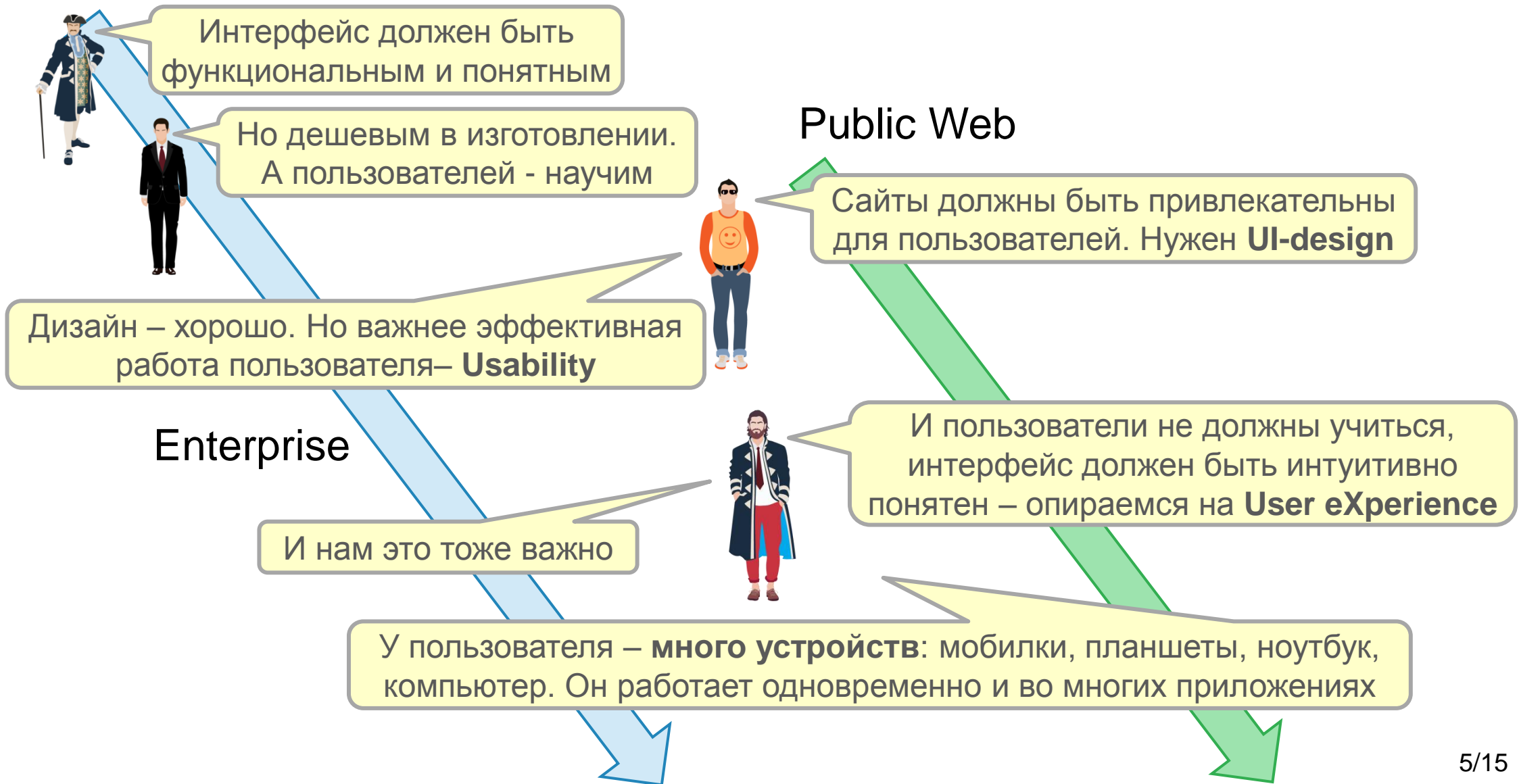
Эпоха НИОКР: делаем правильную систему

Эпоха RUP: делаем систему правильно

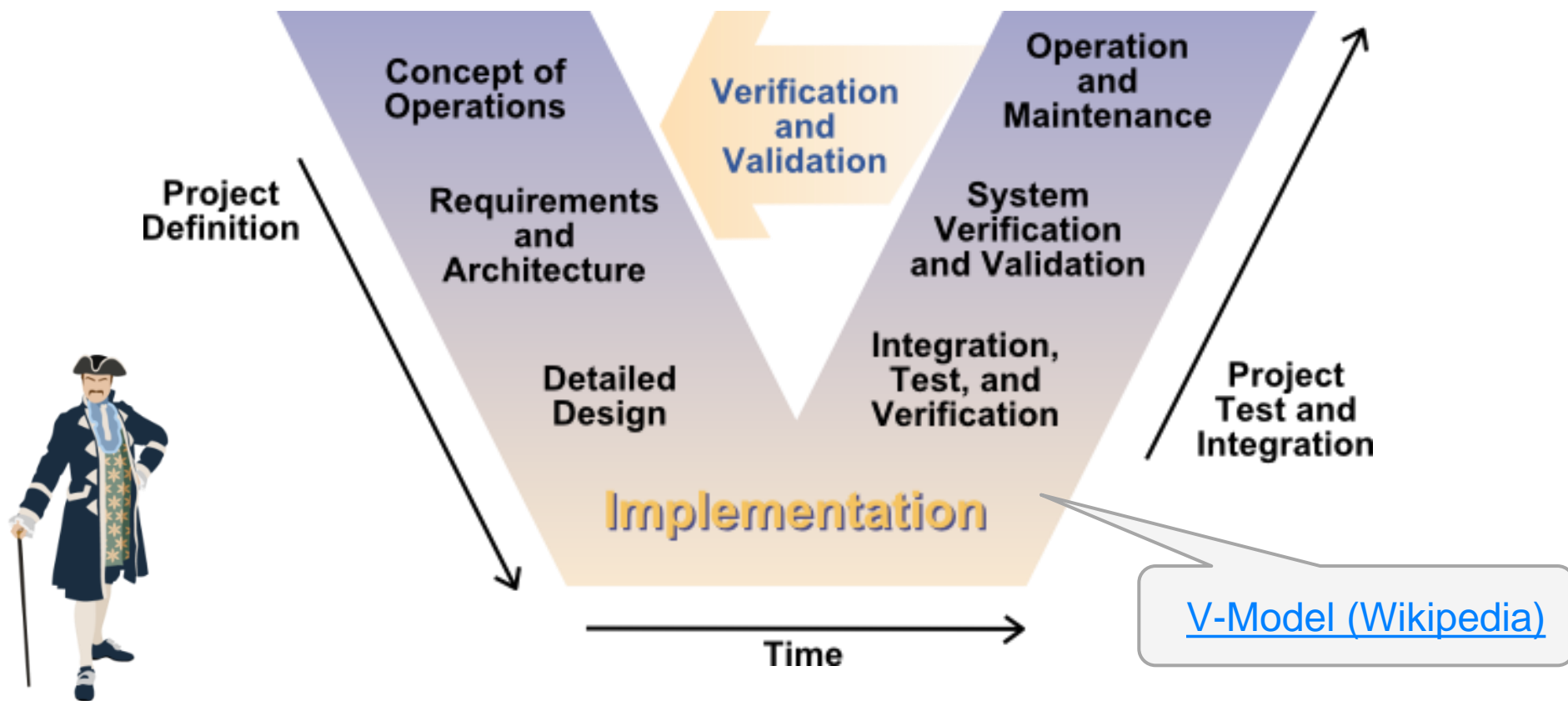
Время Scrum: движемся к цели

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

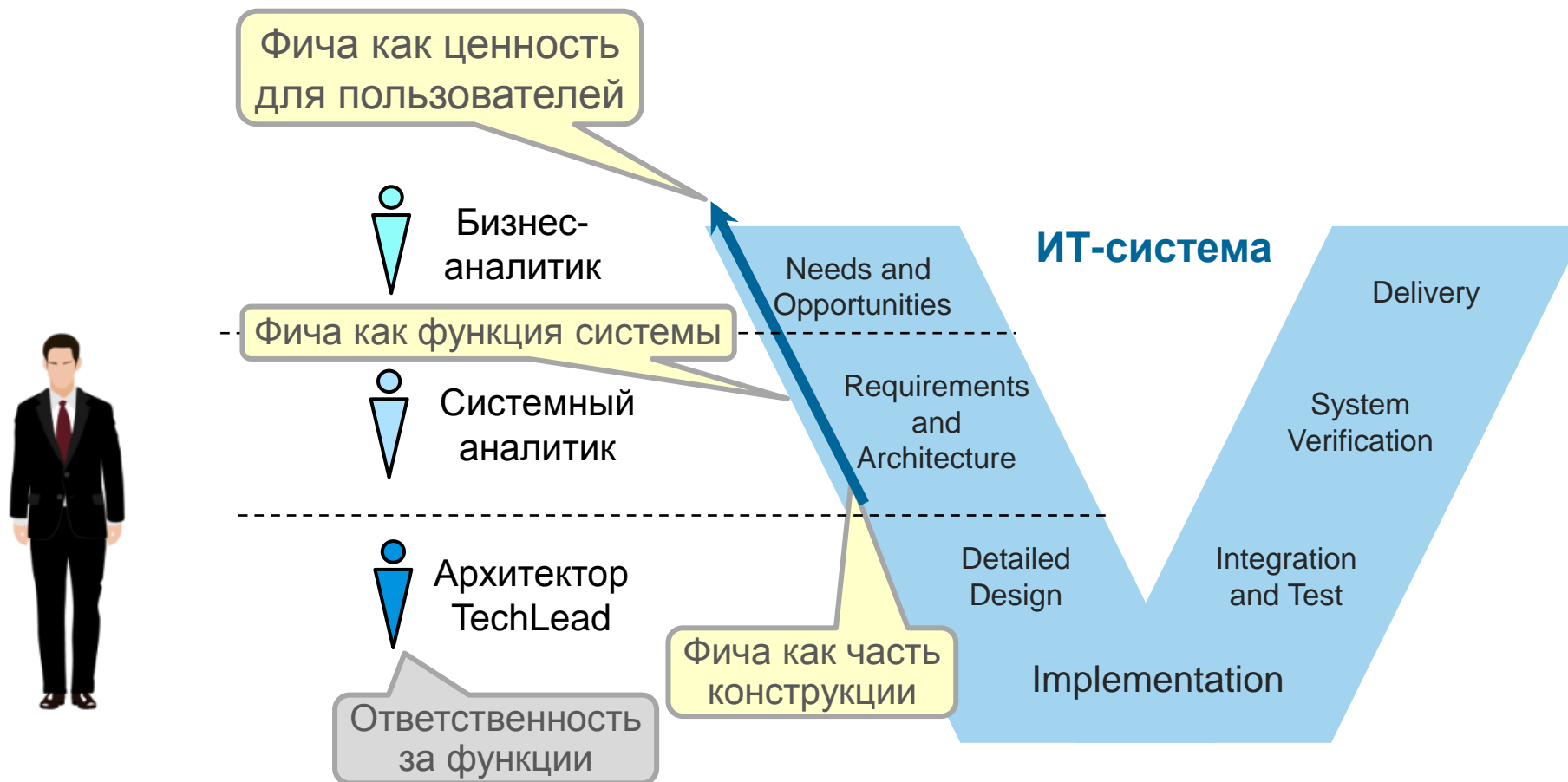
# Эволюция представлений об интерфейсах



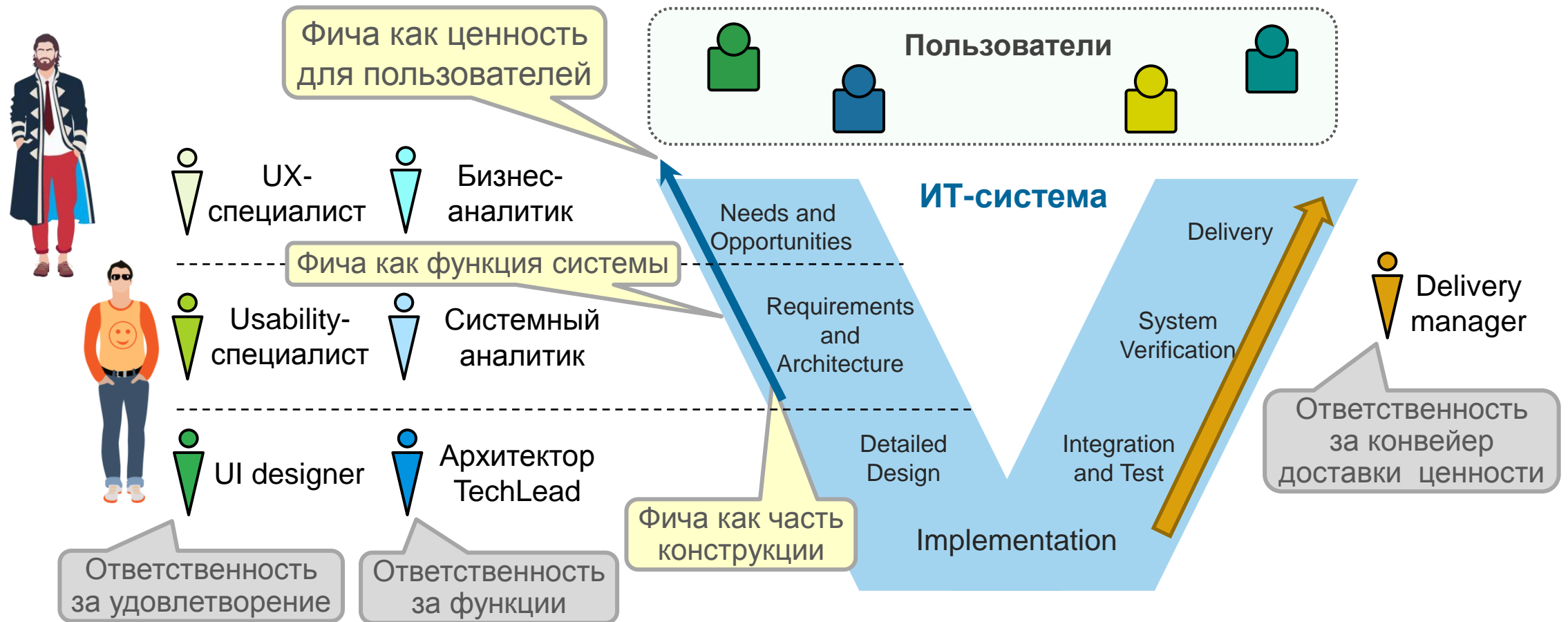
# Представление о проекте – V-модель



# Специализации в проекте



# Удовлетворить пользователей





Удобство во многих приложениях

# Программы для здоровья

- Обычный подход: продукт фокусирован на теме, методике, приборе (сон, фитнес, сердце, браслет) и стремиться «дать все что нужно»
- Развитие: программа – не единственная
  - Агрегаторы (Google Fit) собирают данные
  - Специализация: аналитические программы, им нужно много разных данных
- Потребность человека – собирать свой фреймворк здоровья
  - Собирать данные о себе с разных устройств и с помощью разных программ
  - Не терять накопленную историю, если поменял устройство или подписку
  - Пробовать разные программы для одного или разных аспектов здоровья
- К сожалению, бизнес-модель заточена на эксклюзив...

# Enterprise – мобильные АРМ для магазина

- Сотрудники работают с обычными и специальными гаджетами: смартфон, умные часы, носимый принтер этикеток и сканер, планшет
- Типовые устройства – личные, специальные – компании
  - Использование личных устройств снижает затраты на админов и содержание
  - А стоимость покупки можно компенсировать
- Задача может быть распределена на несколько устройств разных типов
- Софт пишут разные команды, воля к сотрудничеству есть, но шаблонов проектирования – пока нет

# Use case – сквозь много приложений

- Пример: отбор товара по для интернет-заказа в магазине
  - Оповещение о новых приходит на личные часы – они всегда с собой
  - Посмотреть задачу надо на уже смартфоне – фото товара, сканер этикеток
  - Сотрудник собирает товары в коробку, фиксирует это смартфоном
  - В коробку надо положить бумагу с информацией о заказе – со смартфона дать команду, чтобы сервер сделал документ напечатал на ближайшем принтере
  - А если товар не найден, то может потребоваться приложение на десктопе с большим экраном
- Еще пример: обновление ценников при смене цен
  - Тут еще нужен специальный принтер этикеток
  - А еще, если появились покупатели – задачу надо прервать
  - А другой сотрудник через некоторое время – продолжит с прерванного места

# Первым был Microsoft – но не вышло...

- Вместе с Windows8 MS создал концепт будущего Мой отчет [MS DevCon 2013](#)
  - Единый интерфейс на десктопе, планшете и телефоне с прозрачным переходом
  - Взаимодействие приложений на клиенте и в облаке, распределенные облака
  - Но все это – только в собственной экосистеме, а сил на нее не хватило...
- В мире Android и Apple между разными приложениями и устройствами – пропасть
  - Пример: нет легкого инфраструктурного решения, чтобы с мобилки отправить сформировать документ и напечатать на ближайшем принтере
  - На Android среда собирается из кусочков, их надо подгонять (как и в мире Java)
  - Взаимодействие приложений на Apple – только через базу данных или облако
  - А еще Apple для enterprise размывает бренд: iPhone и iWatch не для продавцов
- Но развитие – не остановить: программисты – работают 😊

# То же происходит и в жизни – образование

- Раньше человек шел в ВУЗ, который давал полный курс образования
- Сейчас – много кратких курсов, online, offline и гибридных
- ВУЗы в старой системе по-прежнему пробуют собрать все в свой курс
- А человеку – не нужно, нужна свобода интеграции и курсы по запросу



Три процесса, в которых виден одинаковый характер изменений помогают понять внутреннюю логику.  
Поэтому этот пример приведен вместе с примерами из IT

# Подводя итоги

- Помните историю, и соотносите учебники с контекстом их времени
- По современным задачам – точно нет ни учебников, ни тренингов
- Хотя Вы разрабатываете приложение, думать об удобстве жизни
- Трассируйте Ваши решения до сценариев жизни
- Это – часть системного подхода – думать о месте в надсистеме



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](#).