

Омск. 7-8 декабря 2013

Роли в проекте: как поделить поляну ответственности

Максим Цепков

www.facebook.com/mtsepkov

www.custis.ru



О чем этот доклад?

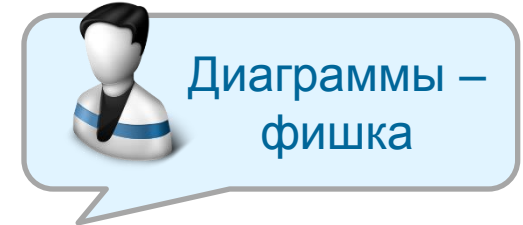
- Процесс разработки и разделение ролей:
 - Классика – водопад и RUP, разделение ролей – оттуда.
 - IT-отрасль меняется, меняются и модели.
- Agile:
 - качнул модель до полной кроссфункциональности
 - потом специализации вернулись
- Тренд – разделение, настроенное на проект



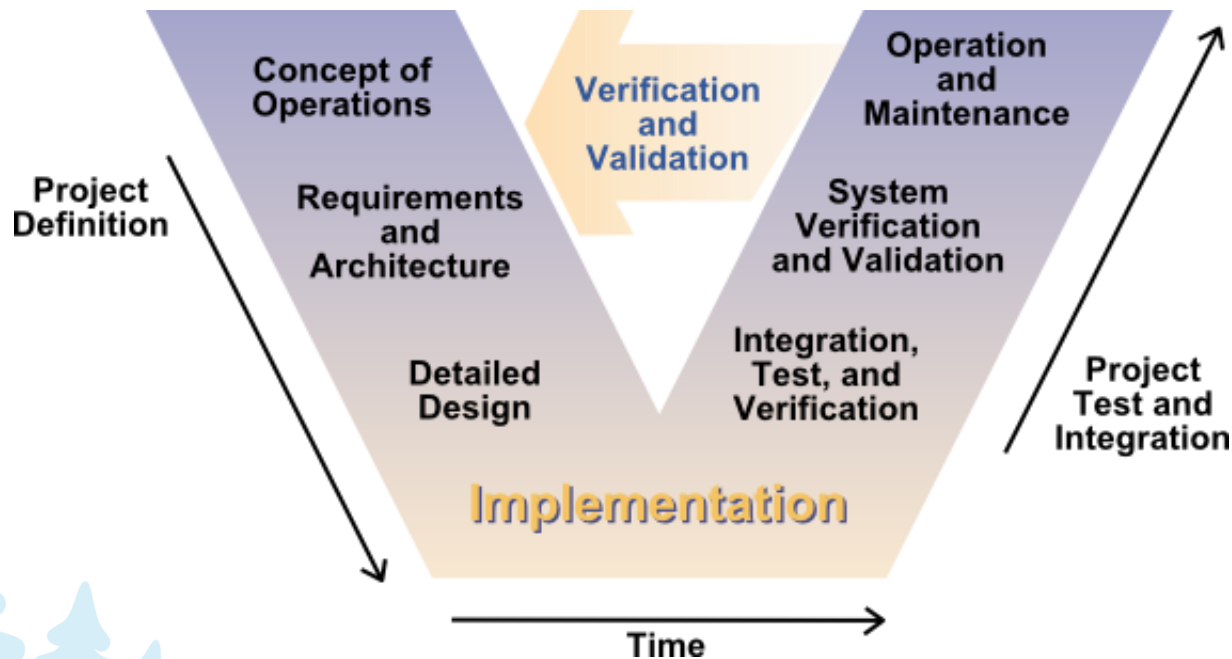
Одного правильного способа – нет
Но многим приятно думать *иначе*

Визуальное представление ролей

- Для эффективного проектирования нужно графическое представление.



- Это оказалось удобно делать на схеме V-модели.



[http://en.wikipedia.org/wiki/V-Model_\(software_development\)](http://en.wikipedia.org/wiki/V-Model_(software_development))

Процесс разработки и роли



Agile – мировой тренд

Это просто факт

➤ Наблюдаемые признаки:

- Признание и использование Agile в ведущих IT-компаниях и в inhouse-разработке.
- Явное упоминание Agile в базовых документах ([SWEBOK](#), [PMBOK](#)).
- Россия – в русле мирового развития.

➤ Мечта о едином, эталонном процессе похоронена.

- Даже в варианте «возьмите только нужное» (PMBOK).

➤ Делаем процесс, адекватный проекту и компании!

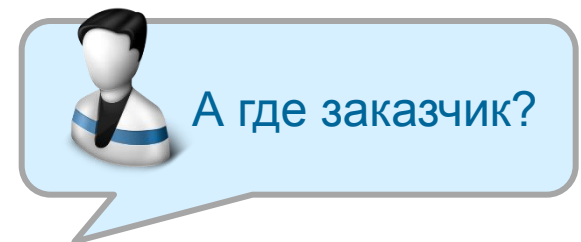
- SCRUM/Canban/XP – лишь распространенные комбинации.
- Комбинируем известные успешные практики, придумываем свои.
- Фокус на эффективные коммуникации и автономность команды.

➤ Essence от [SEMAT](#) – новый уровень

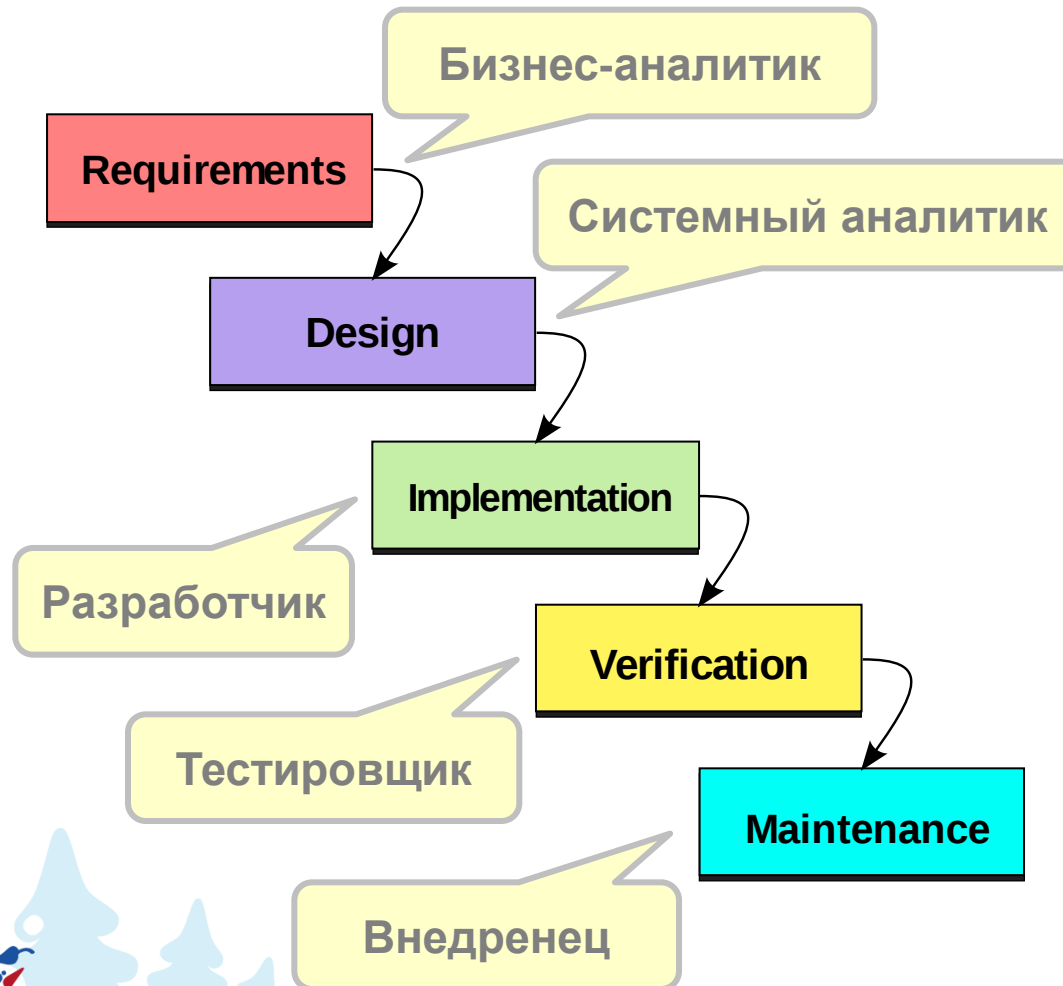


Водопад ушел – роли остались

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

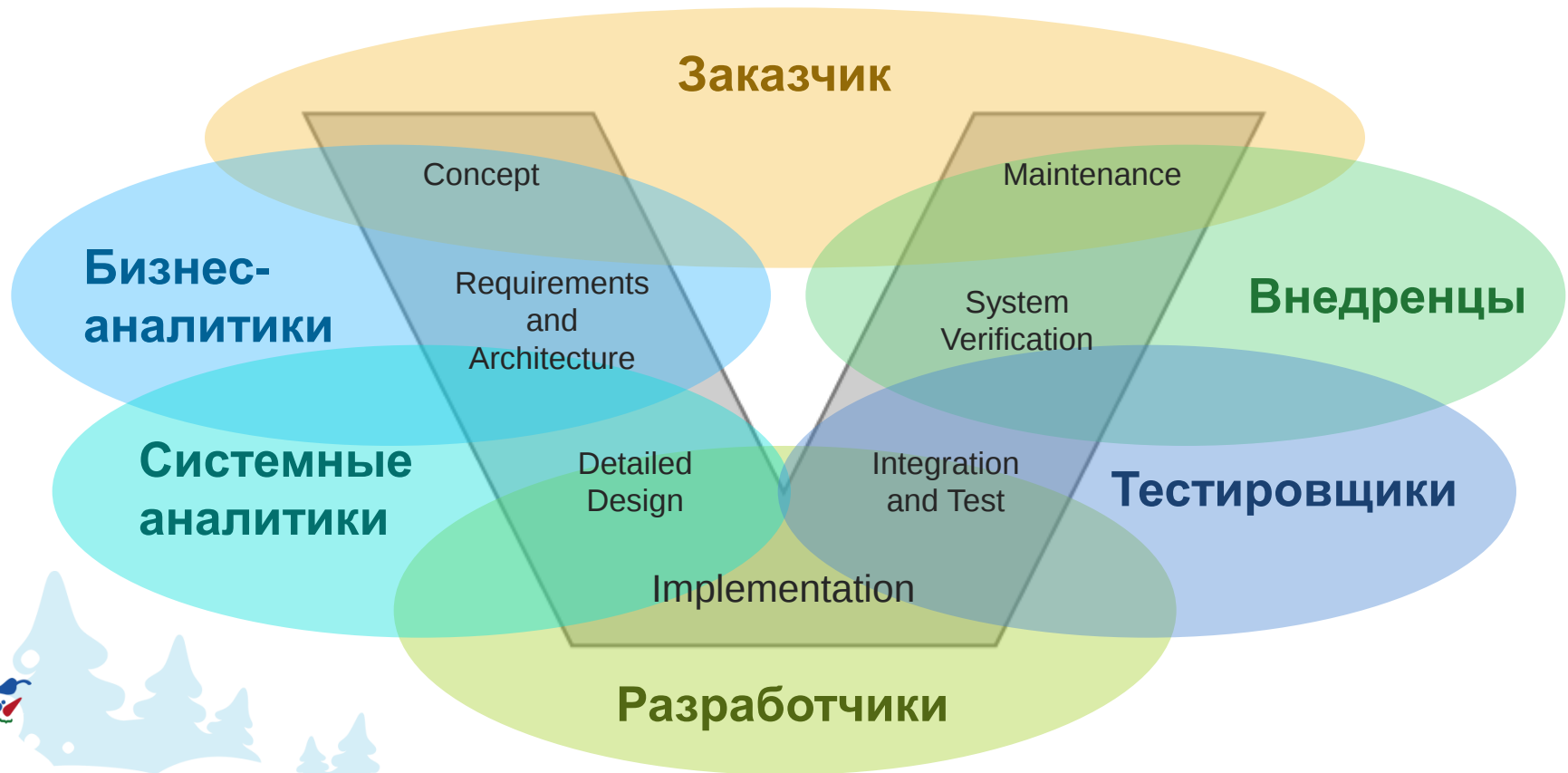


Модель водопада –
http://en.wikipedia.org/wiki/Waterfall_model

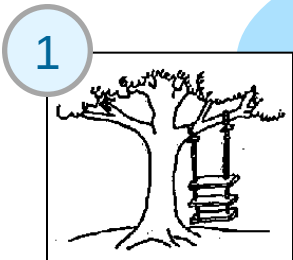
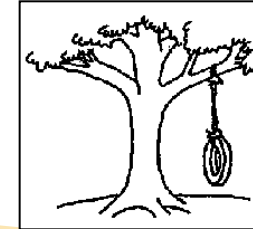
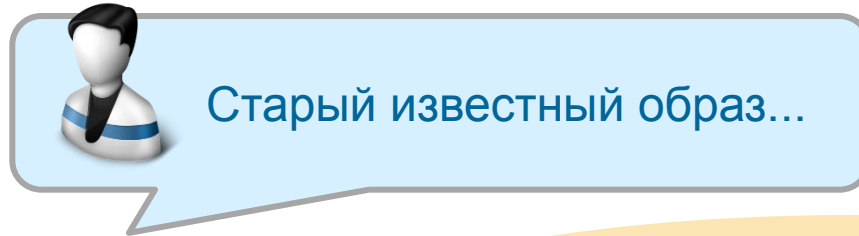


Роли водопада на V-модели

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

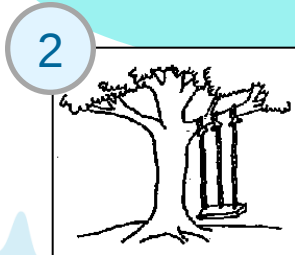


Изменение видения проекта...



Бизнес-аналитики

Системные аналитики



Заказчик

Concept

Requirements
and
Architecture

Detailed
Design

Implementation

Разработчики

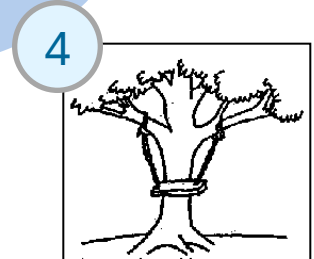
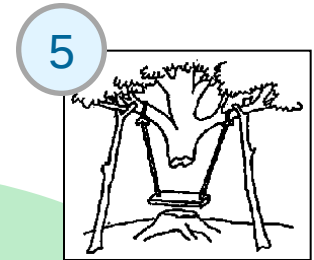
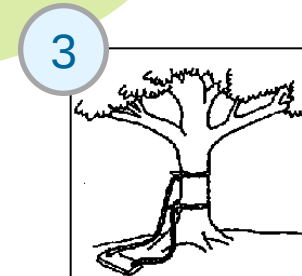
Maintenance

System
Verification

Integration
and Test

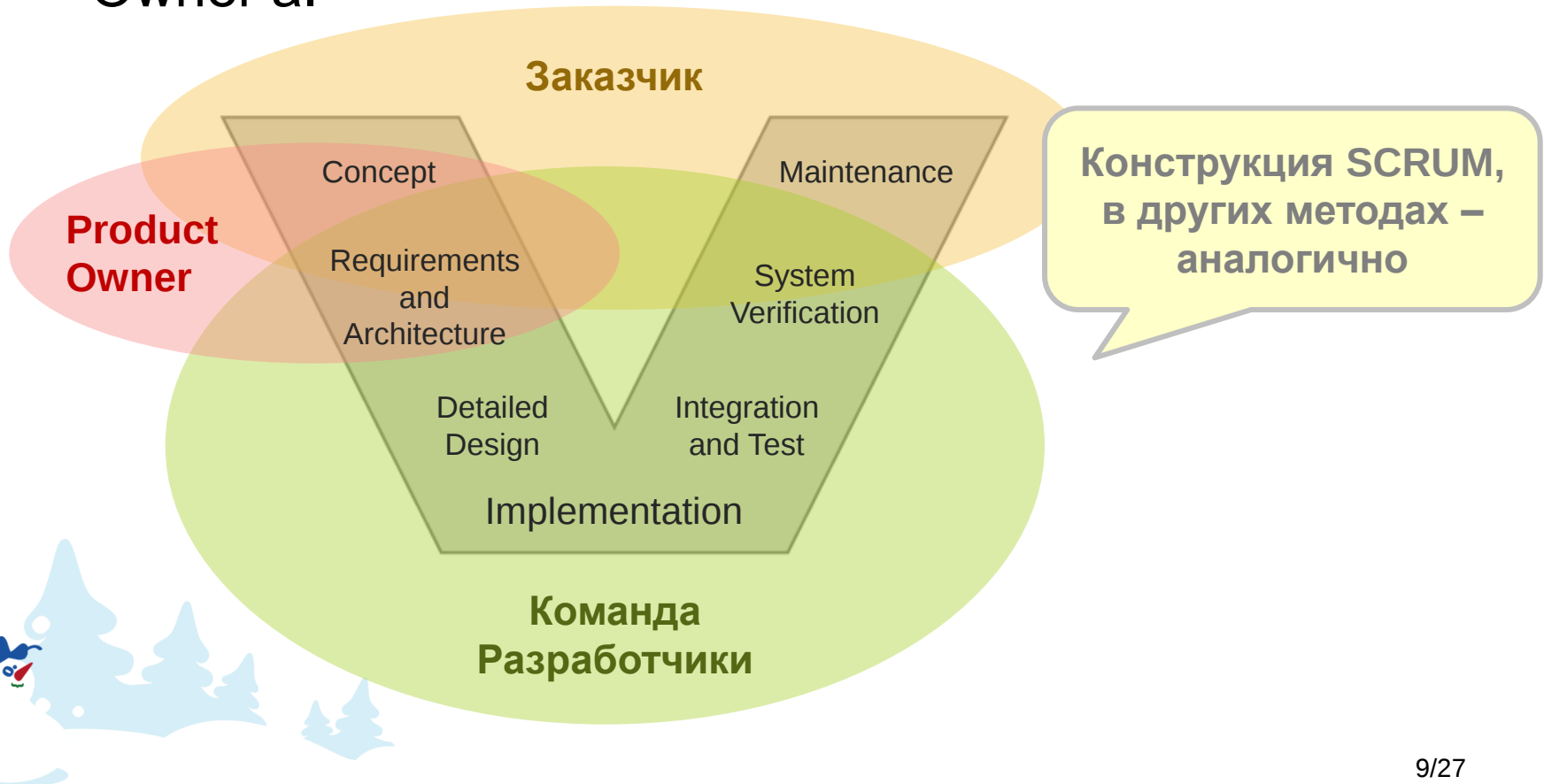
Тестировщики

Внедренцы



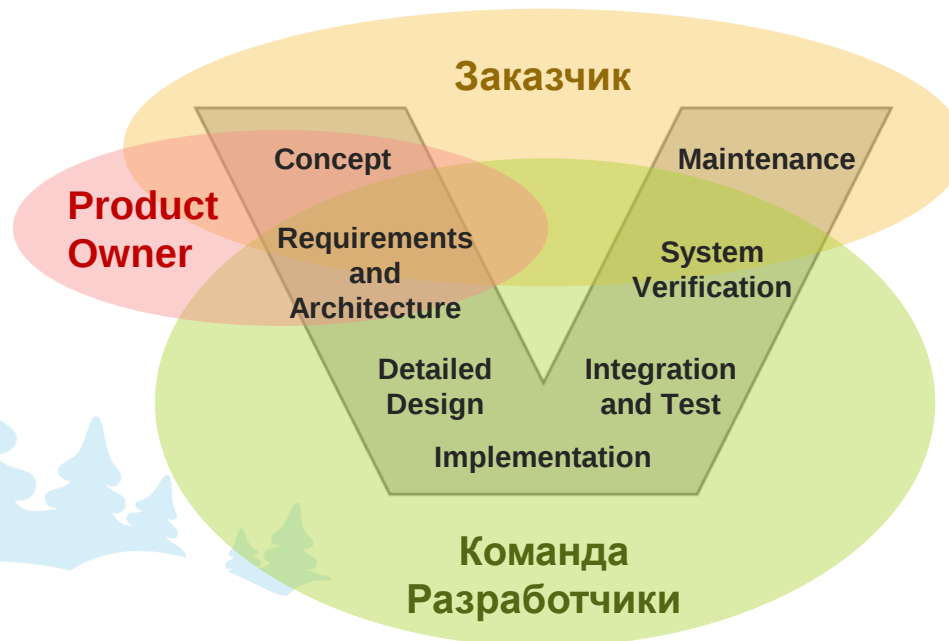
Что предлагает Agile?

- Кросс-функциональная команда разработчиков.
- Взаимодействие с заказчиком через Product Owner'a.



Плюсы и минусы

- 😊 Эффективные коммуникации.
- 😊 Возможность быстрой обратной связи.
- 😞 Большая нагрузка на Product Owner'а.
- 😞 Расширение зоны ответственности Заказчика.
- 😞 Слишком разнообразная работа членов команды.





Happy**Dev**'13

Больше чем конференция!

CUSTIS®

Ищем хорошие варианты

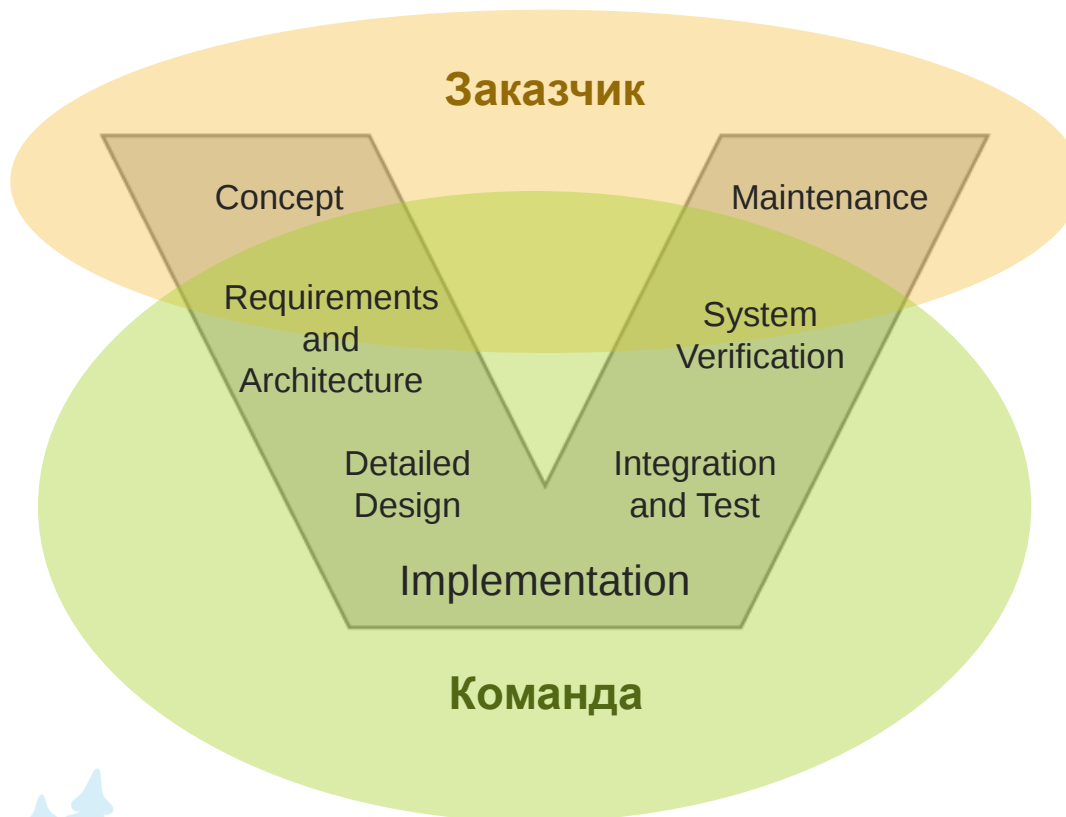


В духе Agile



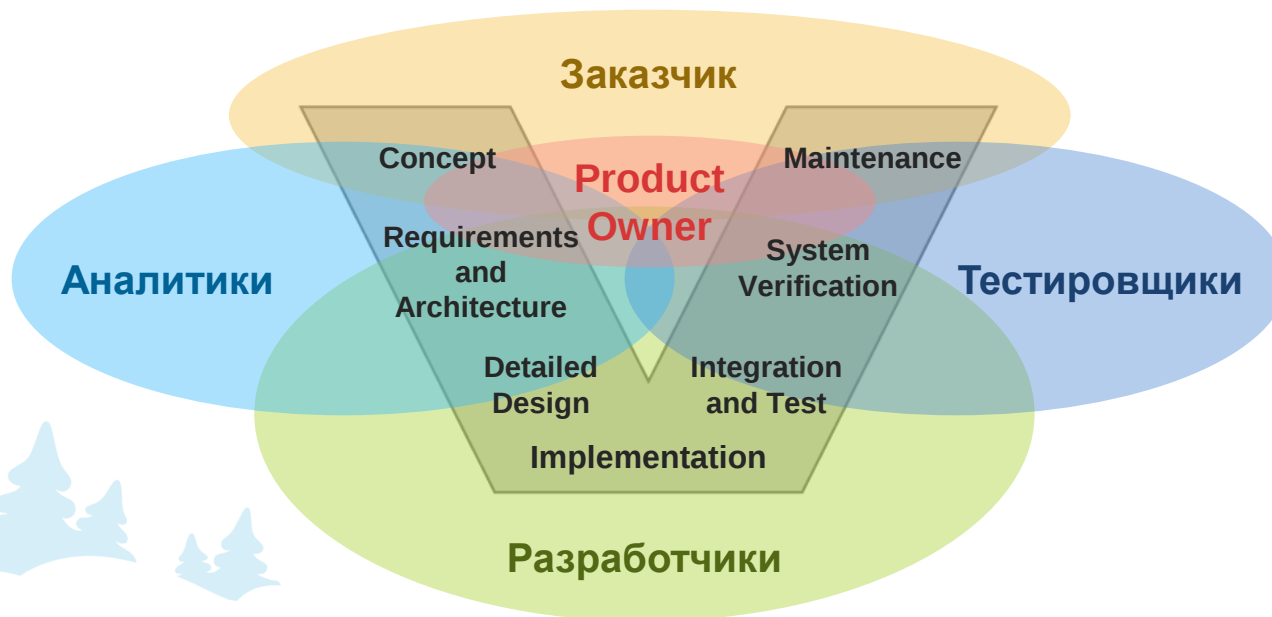
Заказчик – команда

Зона неопределенности – работа с требованиями



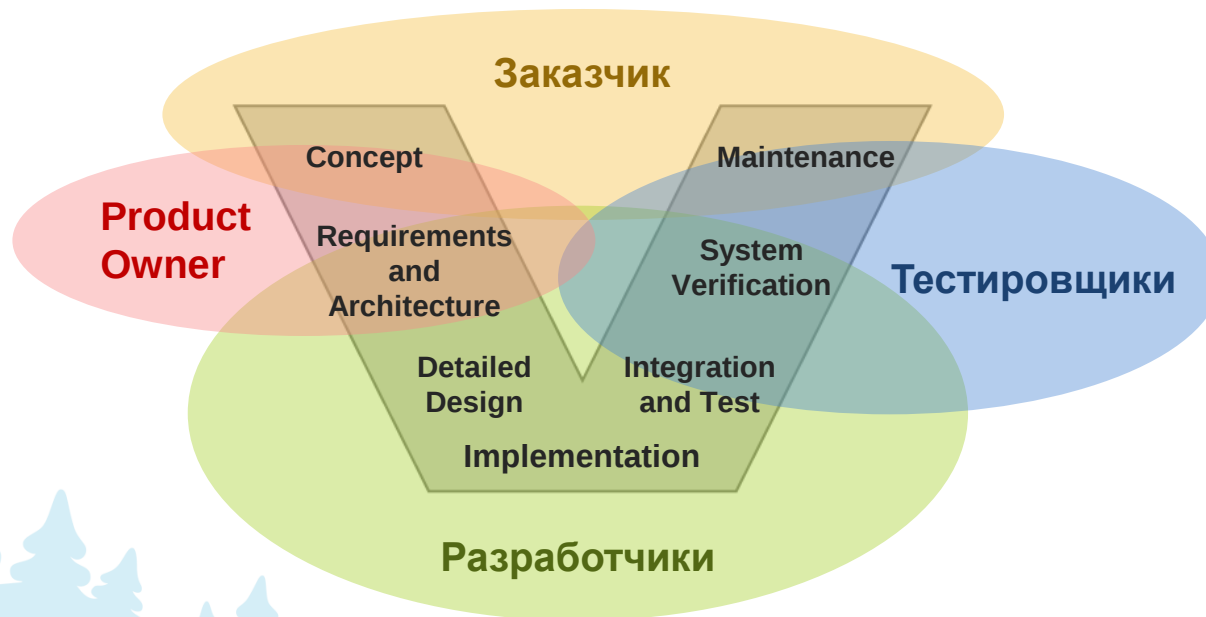
Специализация внутри команды

- Кросс-функциональная команда не означает полной взаимозаменяемости, возможна специализация.
- 😊 Снижается нагрузка на Product Owner'а.
- 😞 Большое число ролей затрудняет коммуникации.
- 😞 Неравномерная нагрузка на роли в ходе проекта.



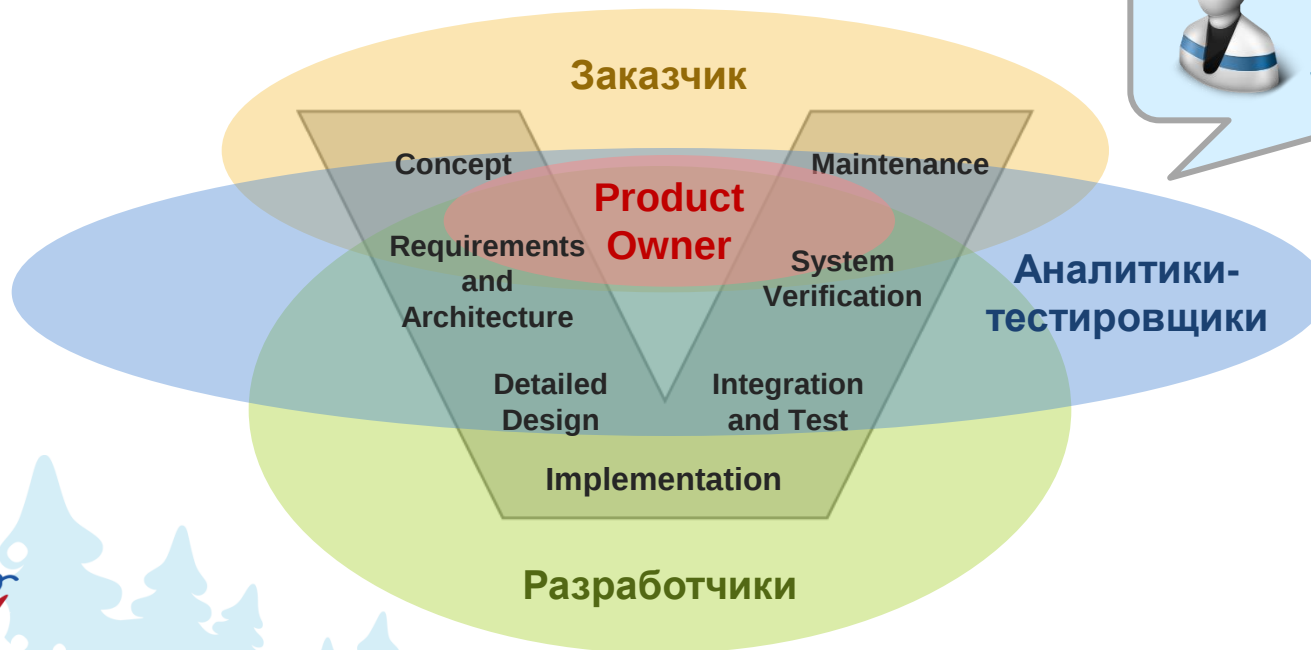
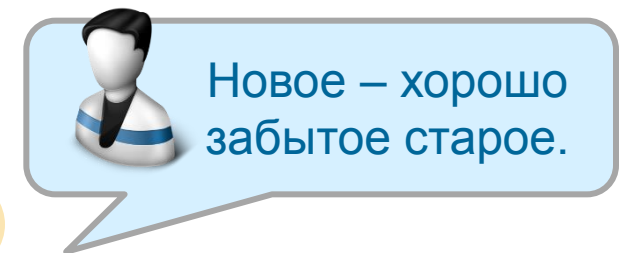
Есть проекты, где аналитики мало

- ▶ Команда разработчиков и тестировщиков – распространенный вариант.
- 😊 Две роли – не много, но достаточно.
- 😞 Не подходит, когда аналитической работы много.



Модель внутреннего заказчика

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



Область применения модели

- Для проектов с полным набором активностей.
- Заказная разработка на полном жизненном цикле: обследование, постановка, разработка, внедрение, **развитие**.
- Для продуктовой разработки тоже применима.
- Модель распространена в мире [Пауль Тернер на Req-Labs](#).



Преимущества модели

- ☺ 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.



Все вместе дает высокое качество услуги для заказчика, особенно при длительном развитии системы.

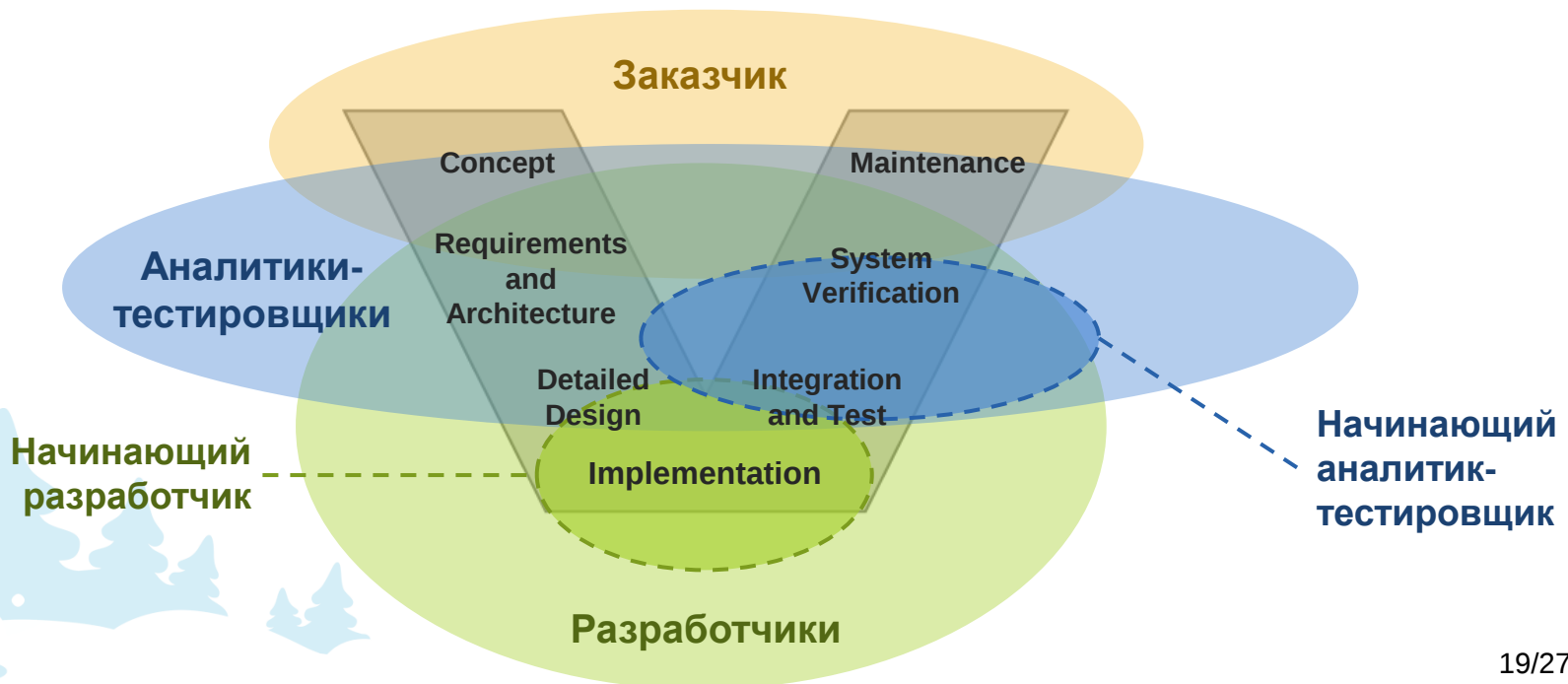
Опыт CUSTIS – типовая команда*

- Соотношение разработчиков и аналитиков – 2:1.
- 6–7 (4–11) человек: 4 разработчика, 2 аналитика и руководитель проекта (Product Owner).
- Члены команды могут заменять друг друга с учетом специализации. У руководителя тоже есть зам.
- Применение DDD дает единый язык общения.
- Часть разработчиков и аналитиков – начинающие, они растут и набирают опыт.
- По мере роста опытные сотрудники уходят в новые проекты, а новички – приходят.

* Для сложных проектов, развивающихся 3–10 лет после внедрения.

Рост новичков в команде

- Активность аналитика начинается с тестирования: освоение системы и бизнес-области.
- Активность разработчика начинается с реализации по проработанным постановкам.
- Постепенно области расширяются...





Кроме разработки
есть другие активности



Менеджерские активности

- Менеджер
- Scrum Master
- Команда
- HR
- ...



Работа с технологиями

- Разработчики
- Тех.лид
- Архитектор
- Product Owner
- ...



Роли и компетенции



SFIA

Skills Framework for Information Age sfia.org.uk

- 90 компетенций в 6 категориях и 20 подкатегориях
- 7 сквозных уровней, характеризующихся по autonomy, influence, business skill: follow, assist, apply, enable, advise, initiate, strategy
- Образование договаривается с бизнесом
- Использование: professional skill + behavioural skill + knowledge + experience + qualifications



Заключение



Подводя итоги

- Создавайте разделение ролей исходя из проекта
- Для визуализации хорошо использовать V-модель
- Эффективные коммуникации – необходимы
- Разделение ролей определяет компетенции
- Учитывайте какие специалисты есть на рынке



Омск. 7-8 декабря 2013

Спасибо!

Вопросы?

Другие доклады lib.custis.ru/MaksTsepkov

Максим Цепков www.facebook.com/mtsepkov

