

# Обзор методологии Scrum

Auriga Inc.  
Дмитрий Сидоренко

# Содержание

- Преимущества
- Происхождение
- Основы методологии
- Роли
- Сопутствующие методологии

# Зачем меняться?

- Существующие методологии плохо приспособлены к изменению требований
- Необходимо знать все требования в начале
- Длительные циклы разработки — проблемы при сдаче
- Требования – абстракция, которая интерпретируется по-разному
- Высокая вовлеченность клиента в начале проекта сходит на нет к окончанию работ
- Недостаточное тестирование
- Проблемы появляются в конце
- Прогресс определяется % от задачи

# Преимущества Scrum

- Прозрачность для бизнеса
- Заказчик может вносить изменения
- Проблемы быстро идентифицируются
- Разработчики вовлечены в процесс
- Результаты быстро доступны для проверки
- Менеджмент видит прогресс
- Менеджмент разгружается
- Прогресс определяется наличием работающего приложения

# Скрам – не панацея

- Проблемы, которые мы решаем, не связаны с процессами, они в людях
- Скрам и Agile основаны на теории, что для разработки программного обеспечения не существует мета-решения. Только framework, который мы изучаем и адаптируем
- Разочаровывающе для тех, кто ищет процедуры и окончательные ответы

# Scrum за 2 минуты

- Scrum – это гибкая методология, которая фокусируется на business value
- Позволяет быстро и последовательно предоставлять работающие части проекта заказчику
- Каждые две недели любой заинтересованный человек может участвовать на показе текущей версии
- Заказчик задает приоритеты. Команда самоопределяется, чтобы производить наиболее важную для заказчика функциональность
- Scrum задает только общие правила управления проектом

# Agile Manifesto

[www.agilemanifesto.org](http://www.agilemanifesto.org)

- Люди и общение, а не процессы и инструменты
- Работающее приложение, а не сложная документация
- Сотрудничество с клиентом, а не составление контрактов
- Реакция на изменения, а не следование плану

# Что значит “Гибкая”?

- “Гибкость – означает быть открытым относительно того, что ты можешь сделать и делать это” Кент Бек

+

+

+



# Происхождение

- Scrum – команда в регби
- “The New New Product Development Game”, Harvard Business Review, 1986, Takeuchi and Nonaka
- Origins of Scrum
  - <http://www.agilealliance.org/system/article/file/786/file.pdf>

# Компании

- Microsoft
- Yahoo
- Google
- Electronic Arts
- High Moon Studios
- Lockheed Martin
- Philips
- Siemens
- Nokia
- Capital One
- BBC
- Intuit
- Time Warner
- Nival
- Luxoft

# Характеристики

- Самоопределяющаяся команда
- Продукт разрабатывается в процессе серии итераций (sprints)
- Требования записываются в “product backlog”
- Инженерные практики не являются частью Scrum
- Использует простые правила для создания гибкой среды разработки проектов
- Один из “agile” процессов

# Scrum



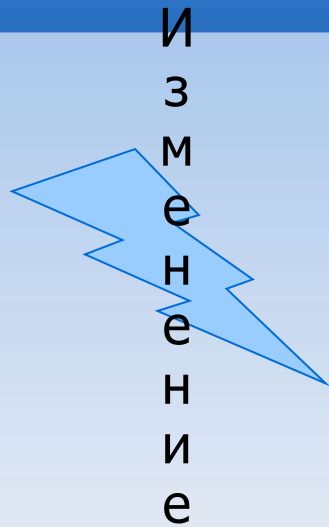
Потенциально  
готовый  
к поставкам  
продукт

Product  
backlog

# Sprints

- Проект разрабатывается в серии спринтов
- Типичная продолжительность – от 2-х недель до месяца
- Жесткое ограничение по времени
- Постоянная продолжительность спринта привносит ритм в разработку
- Продукт проектируется, кодируется и тестируется на протяжении одного спринта
- В конце спринта – полностью готовая функциональность

# Изменения во время спринта



- Планируйте длительность спринта исходя из соображения о том, как долго вы можете работать, не внося изменения в план работ

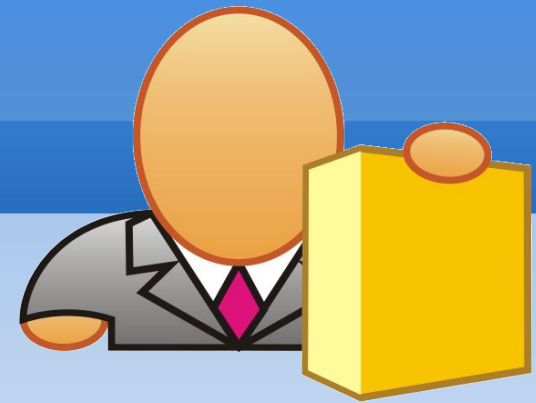
# Framework

# Роли

- Нет фиксированных позиций
- Все участники кроссфункциональны
- Плоская структура
- Реальная жизнь вносит коррективы



# Product owner



- Один человек
- Определяет требования (vision)
- Определяет дату релиза и наполненность
- Ответственен за доходность проекта (ROI)
- Приоритизирует требования, исходя из их рыночной ценности
- Корректирует приоритеты на каждой итерации, если необходимо
- Постоянно общается с всей командой
- Принимает работу

# Как найти хорошего РО

- Хорошим Product Owner'ом не рождаются
- Эксперт в бизнес домене, готовый потратить 30 минут в день на общение с командой
- Product Owner заинтересован в проекте
- Высокопоставленный чиновник – редко хороший РО
- Вносите практики постепенно

# Занятость РО

- Полдня на планировании спринта
- 15-30 минут в день
- 2 часа на спринт-ревью
- Несколько дней на начальную идентификацию User Stories
- Желательна доступность в режиме онлайн
  - skype, icq, messenger

# ScrumMaster



- Ответственен за внедрение практик
- Устраняет препятствия
- Ответственен за эффективность работы команды
- Защищает команду от внешних воздействий
- Не раздает задания
- Обеспечивает видимость и прозрачность

# Кто такой Скрам Мастер - 2

- Лидер и помощник
- Ответственен за
  - удаление препятствий
  - обучение клиента
  - упрощение жизни команды
  - улучшение производительности команды
  - улучшение применяемых инженерных практик

# Памятка Скрам Мастера

- Command & control – иллюзия
- Магии не существует
- Прозрачность процессов

# Команда

- Обычно 5-9 человек
- Кроссфункциональные члены команды: программисты, тестеры, дизайнеры...
- Полный рабочий день
- Самоопределяющаяся
- В идеале, нет позиций (PM, TL, tester)
- Отвечает за результат перед РО

# Product backlog

- Список желательной функциональности
- Управляет Product owner
- Приоритизируется Product owner
- Реприоритизируется в начале спринта
- В идеале написан так, что каждый элемент описывает Use case конечного пользователя



# Пример

# Что НЕ Скрам?

- Противоречие Agile Manifesto
- Отсутствие итераций
- Отсутствие или игнорирование обратной связи
- Отсутствие пула задач с заданными приоритетами
- Непрозрачность

# Когда Скрам не нужен?

- Проекты делаются полностью, вовремя, в полном объеме
- Команда собирается только на краткосрочный проект

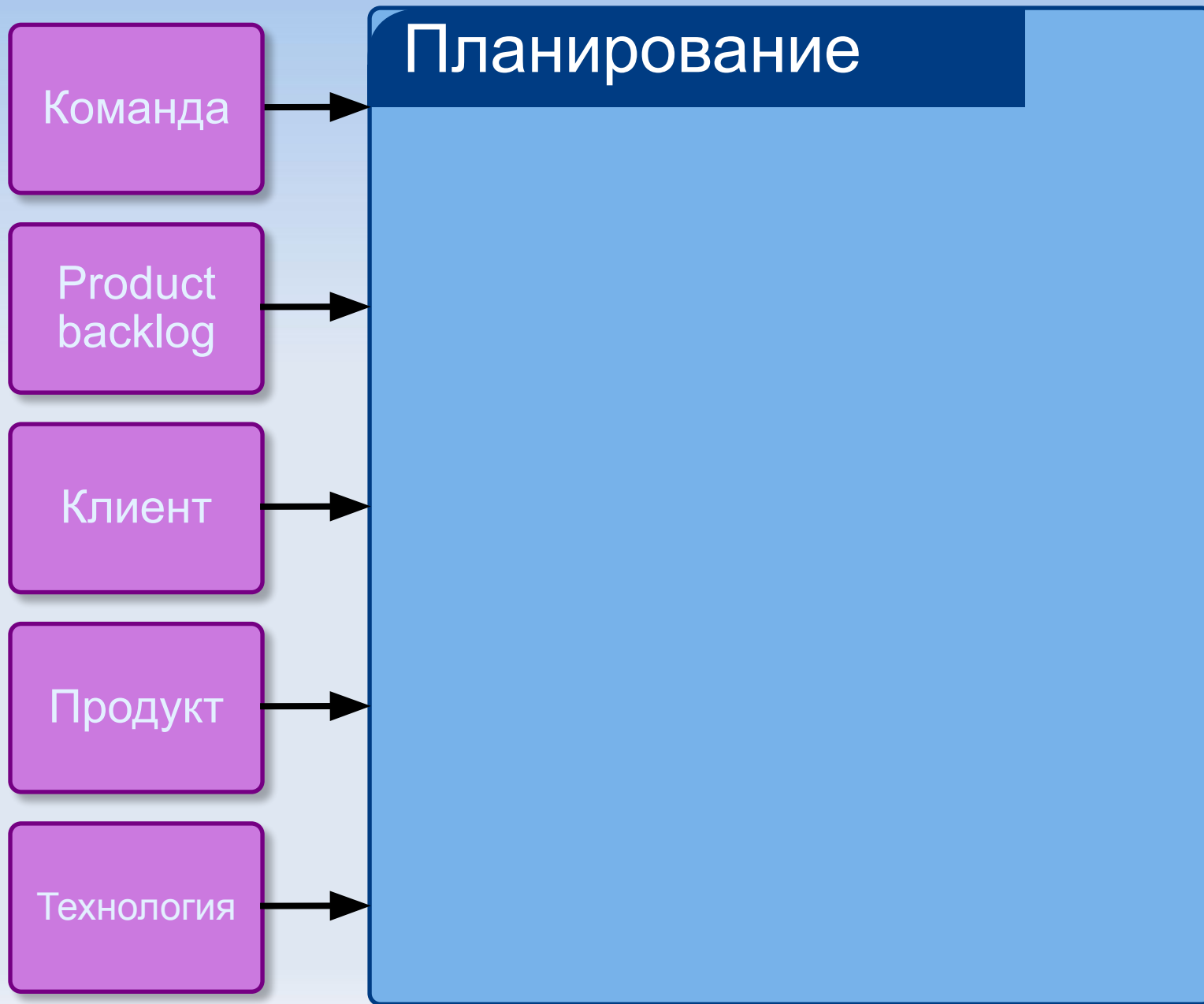
# Когда Скрам не работает?

- Гос. проект
- Тонущий проект, который отдали в офшор
- Скрам Мастер – традиционный ПМ
- Во всех остальных случаях, когда не работают другие методологии
  - текучка
  - распределенность
  - низкий уровень технических знаний
  - ...

# Команда: самоорганизация

- Не происходит сама по себе
- Требуется внешних условий
- Команда должна понимать, зачем организовываться
- Частые и неформальные отзывы о работе очень важны
- Требуется времени
  - 4 этапа становления команды

# Планирование Спринта



# Цель спринта

- Короткое предложение, описывающее, каким должен быть результат спринта

БД

Запустить приложение  
на MS SQL

Интерфейс

Написать графический  
интерфейс

Финансы

Включить поддержку  
загрузки котировок  
в реальном времени

# Планирование спринта


- Скорость работы команды задает объем работ на спринт
- Суммарный объем задач на спринте не должен превышать возможности команды
- Увеличение объема работ неизбежно приводит к падению качества



# Подробнее про планирование

- Команда выбирает, что из product backlog будет реализовано на спринте
- Создается Sprint backlog
  - Задачи идентифицируются и оцениваются
  - Все делается командой, не Scrum master
- Учитывается High-level design

Как отдыхающий,  
я хочу  
посмотреть  
на фото отелей



# Управление sprint backlog

- Работа выбирается самостоятельно, назначений нет
- Постоянная переоценка сложности задач
- Любой член команды имеет доступ к бэклогу спринта
- Изменения во время sprint нежелательны
  - если нужно “очень срочно” - перенести часть задач обратно в product backlog

# Ежедневный Scrum

- Характеристики
  - Ежедневно, в одно время
  - 15 минут
  - Обмен информацией
- Не для решения проблем
- Приглашены все
- Только участники команды могут говорить (product owner – часть команды)
- Ведет ScrumMaster



# Три вопроса

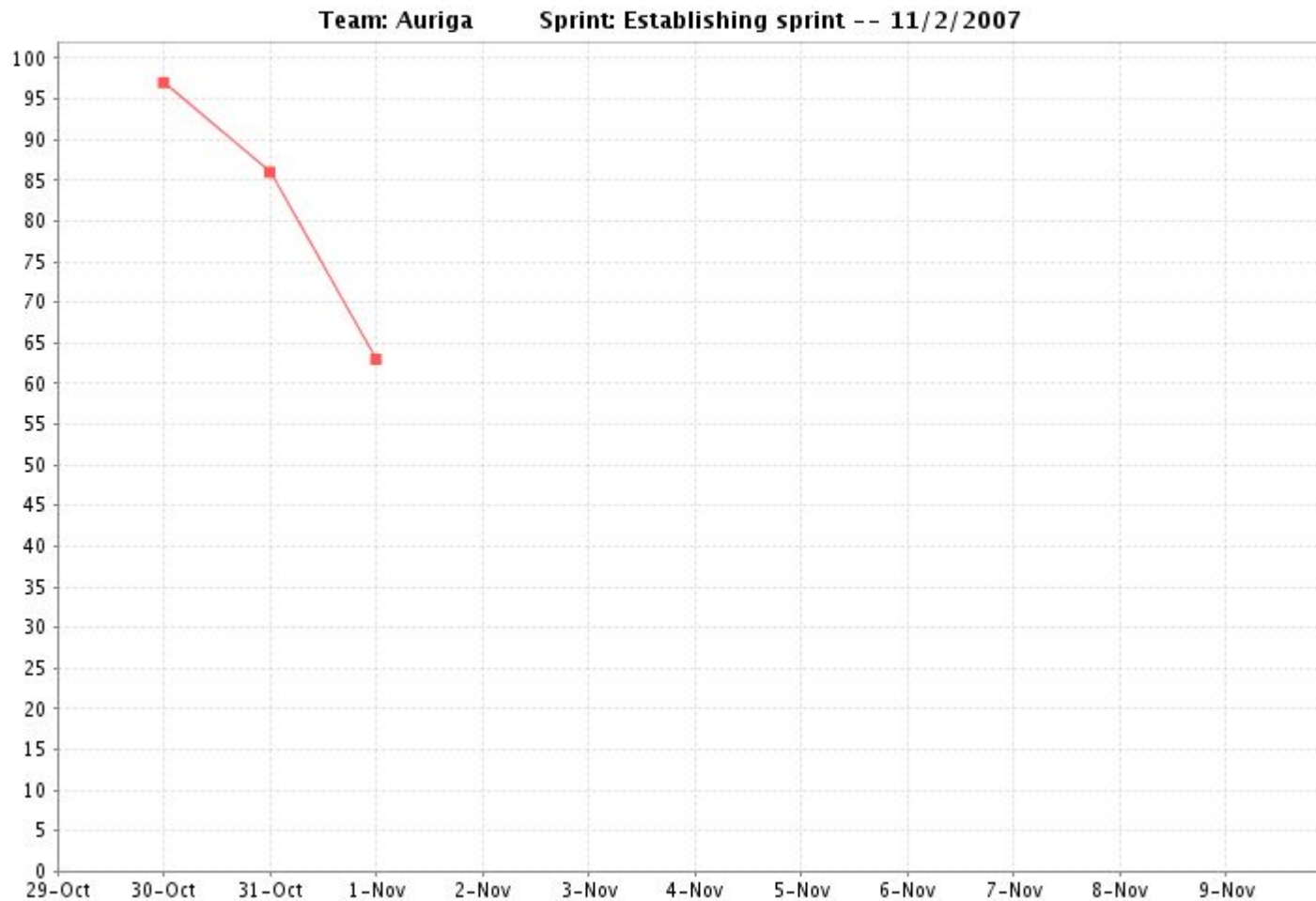
- Это не статусный отчет СкрамМастеру!

# Пример: спринт

## Current Sprint Burndown

For Teams: Auriga

### Auriga



# Спринт ревью

- Команда представляет, что было сделано на спринте
- Фокус на результат, а не процесс
- Обычно принимает форму демонстрации
- Неформально
- 2 часа на подготовку
- Без слайдов
- Вся команда участвует
- Приглашены все

# Ретроспектива

- Пересмотр эффективности практик
- 15-30 минут
- После каждого спринта
- Вся команда участвует
- Возможно, приглашены клиенты

# Инженерные методологии

- Unit testing
- Test Driven Development
- Continuous integration
- Refactoring
- Code review



# Estimation Practices

- User Stories
- Estimation Game

# Пример: Product backlog

## Product Backlog Items

From Releases: Release 6.4

Of Themes: bug, refactor, request, including unthemed

### Release: Release 6.4

Key	PBI Title	Est.	Status	BW	rBV	ROI	Themes	Sprint end [Team]
vstor-30	Adding files problem to main table	4	Done	14	5%	2	bug	11/9/2007 [Auriga]
vstor-2	Script error when trying to provision 2 unsupported content at once	4	Done	16	6%	3	bug	11/9/2007 [Auriga]
vstor-4	When adding file there is no indication that information is required in the 'Name' field	3	Done	16	6%	3	bug	11/9/2007 [Auriga]
vstor-5	No results presented when selecting CC or CP filter with specific initial state	4	Done	18	6%	3	bug	11/9/2007 [Auriga]
vstor-8	Typo in 'Dynamic Counters' section	1	Done	13	5%	8	bug	11/9/2007 [Auriga]
vstor-7	Can't remove several variations simultaneously	8	Done	15	5%	1	request	11/9/2007 [Auriga]
vstor-12	Mechanism of Sorting by 'Provision Time' does not work correctly	4	Done	13	5%	2	bug	11/9/2007 [Auriga]
vstor-9	Item search, replace Go with Search	2	In Progress	16	6%	5	bug	11/9/2007 [Auriga]

# Вариант определения приоритета

- Определение важности User Story
- Effort – затраты на реализацию
- Benefit – преимущество от включения
- Penalty – урон при отсутствии
- Business weight = benefit + penalty
- Release business value =  $BW / \text{SUM}(BW)$
- ROI =  $rBV / \text{Effort} \%$

# User Story

- Высокоуровневое описание функциональности с точки зрения конечного пользователя
- Помогает разработчикам оценивать проект не с технической точки зрения
- Помогает избавиться от “как сделано” в пользу “что сделано”
- Могут разбиваться на более мелкие в процессе работы

# Good User Story

- INVEST
  - Independent
  - Negotiable
  - Valuable
  - Estimatable
  - Sized Appropriately
  - Testable

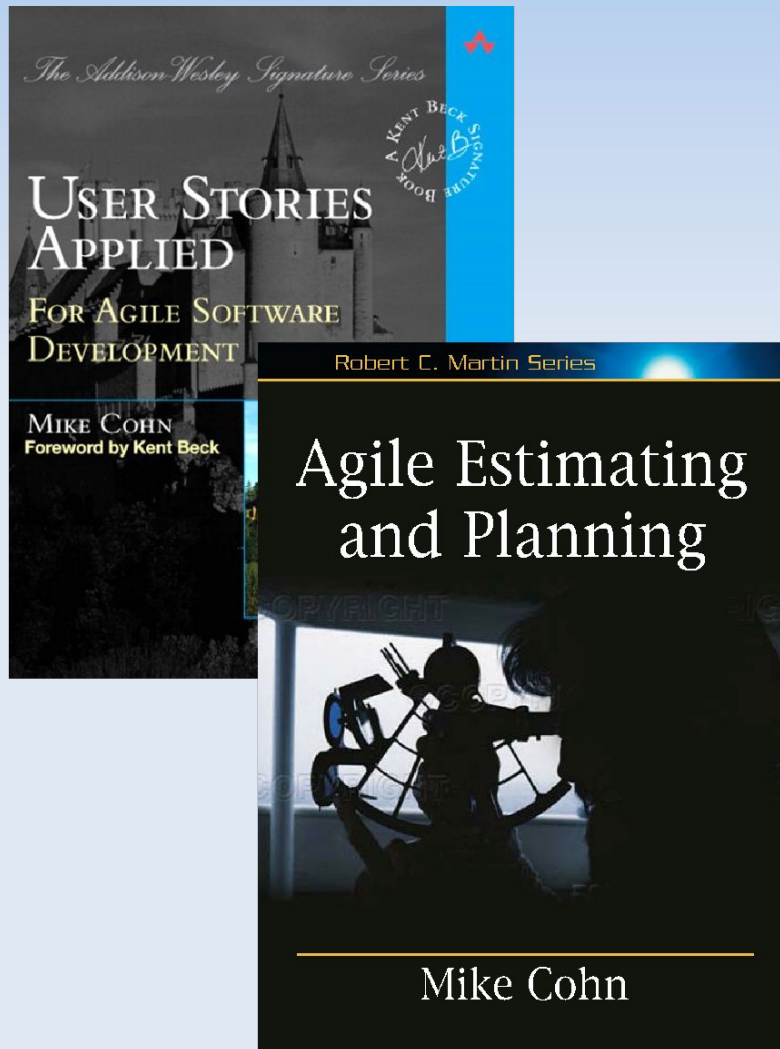
# Где детали?

- Как пользователь, я хочу отменить бронь
  - Полный или частичный возврат денег?
  - Какой лимит во времени?
    - Единый для всех пользователей?
    - Единый для всех отелей?
  - Следует ли слать подтверждение пользователю?

# Estimation Game

- Основана на Expert Estimations
- Вся команда принимает участие
- Оценки даются независимо, результаты сверяются и обсуждаются
- Раунды оценок

# Подробнее об оценке



- Agile Estimating and Planning, Mike Cohn
- User Stories Applied, Mike Cohn



# Изменения в Scrum

- Принципы Scrum — не безусловные истины
- Tailoring допустим и приветствуется
- Вносите новшества в команду постепенно

# Возможные проблемы

- Большие команды
  - Scrum of Scrums
- Клиент требует следования CMMi
  - Scrum возможно сертифицировать по CMMi Level 5
- Нет возможности найти на стороне заказчика РО
  - РО — внутри компании, возможно, не разработчик

# Куда пойти

- Каждые две недели – семинары AgileRussia
- [www.agilerussia.ru](http://www.agilerussia.ru)
- [www.mountangoatsoftware.com/scrum](http://www.mountangoatsoftware.com/scrum)
- [www.scrumalliance.org](http://www.scrumalliance.org)
- [www.controlchaos.com](http://www.controlchaos.com)

# Что читать

- Экстремальное программирование, Кент Бек
- Экстремальное программирование: планирование, Кент Бек и Мартин Фаулер
- Agile Estimating and Planning, Mike Cohn
- Agile Project Management with Scrum, Ken Schwaber
- Agile Retrospectives, Esther Derby and Diana Larsen
- Agile Software Development Ecosystems, Jim Highsmith
- Agile Software Development with Scrum, Ken Schwaber and Mike Beedle
- Scrum and The Enterprise, Ken Schwaber

# Credits

Mountain Goat Software

Mike Cohn

Mike Vizdos

# Контакты

Дмитрий Сидоренко

[dmitry.sidorenko@auriga.ru](mailto:dmitry.sidorenko@auriga.ru)

skype: [dmitry.sidorenko.work](https://www.skype.com/people/dmitry.sidorenko.work)