



IBM Software Group

# Agile-методы и инструменты Rational

Дмитрий Лапыгин  
технический специалист по продуктам IBM Rational  
email: [dmitry\\_lapygin@ru.ibm.com](mailto:dmitry_lapygin@ru.ibm.com)



**Rational.** software

## Что такое Agile?

### *Manifesto for Agile Software Development*

- ▶ Individuals and interactions over processes and tools
- ▶ Working software over comprehensive documentation
- ▶ Customer collaboration over contract negotiation
- ▶ Responding to change over following a plan

### • Основные принципы

- ▶ “Fits just right” процесс, максимально соответствующий потребностям и не содержащий ничего лишнего
- ▶ Непрерывное тестирование
- ▶ Тесное командное взаимодействие
- ▶ Быстрая реакция на изменения
- ▶ Постоянное вовлечение заказчика в общий процесс
- ▶ Как можно более частый выпуск работающего продукта



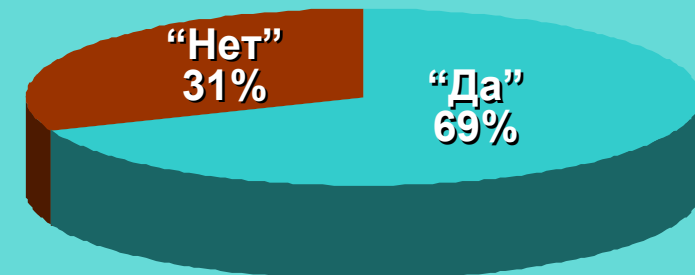
## Agile становится массовым

- 35% участников опроса работают в Agile-проектах (в том числе пилотных)
- только 12 % не видят смысла использовать Agile-методы в своей организации

Таким образом 88 % опрошенных (треть – организации более чем 10,000 сотрудников) используют или рассматривают возможности использования Agile-процессов

- Excerpt from “And the Agile Survey Says...”  
Agile Journal, March 6, 2006

Используете ли Вы Agile-технику?



Source: Ambler ‘Agile Adoption Rate Survey’  
of over 781 Dr. Dobb’s subscribers, March 2007

## Мифы и реальность Agile

- **Отсутствие документации – *Agile-документирование***
- **Отсутствие дисциплины – *высокие требования к дисциплине***
- **Отсутствие планирования – *JIT (just-in-time) планирование***
- **Непредсказуемость – *хорошая предсказуемость***
- **Не масштабируется – *Eclipse как контрпример***
- **Модная причуда – *быстро становится нормой***
- **RUP не является Agile – *RUP настолько Agile, насколько вы это сделаете при адаптации***
- **Нельзя контролировать расходы – *обеспечивается контроль бюджета, расписания и содержания проекта со стороны заказчика***

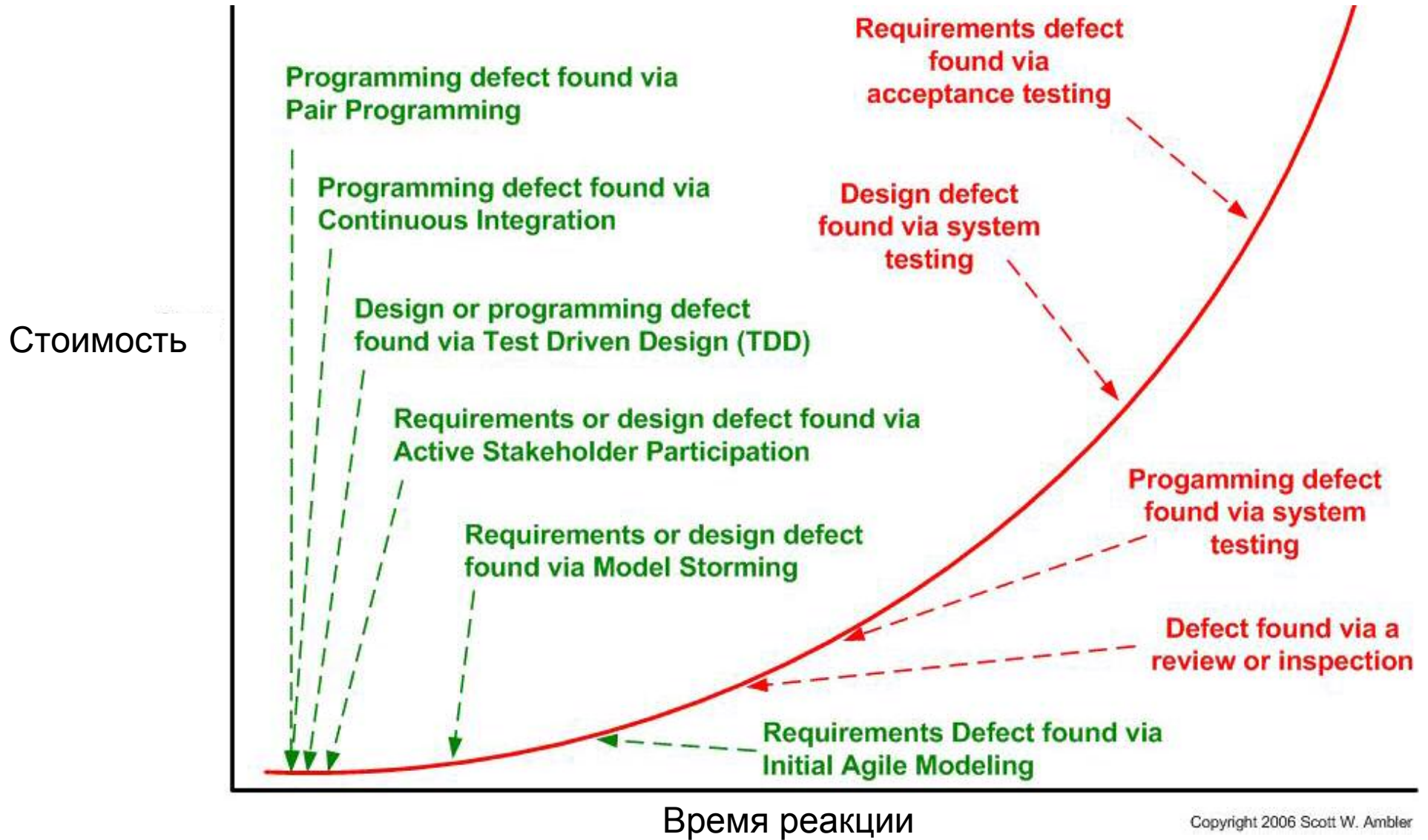


## Особенности Agile-подхода

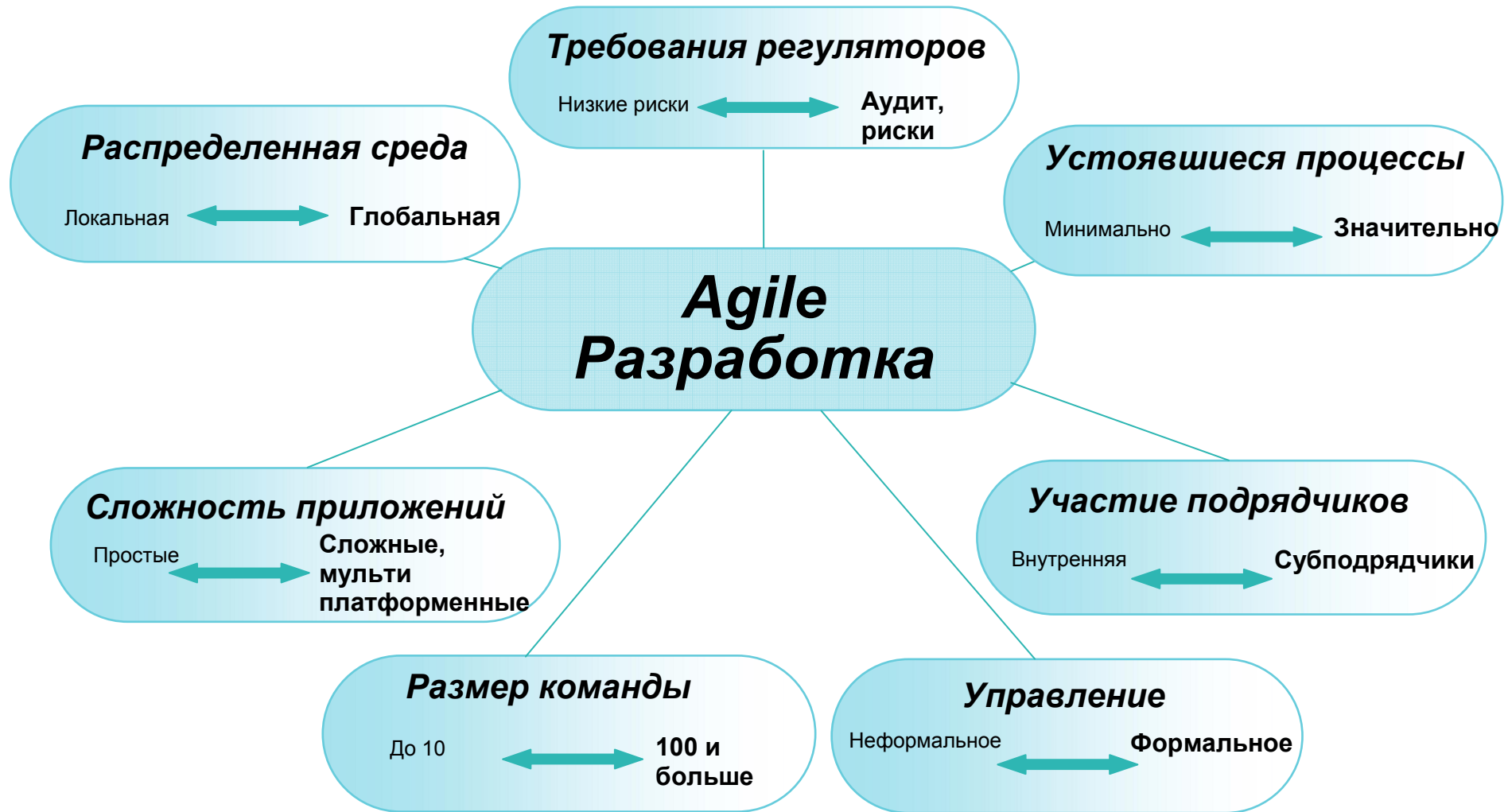
- **Фокус на взаимодействиях**
  - ▶ Меньше бумаг, больше обсуждений
  - ▶ Активное вовлечение заказчиков
- **Фокус на работающем продукте**
  - ▶ Хорошие отзывы упрощают управление Agile-проектами
  - ▶ Требуется меньше документации
  - ▶ Снижение бюрократизации
- **Универсализация специалиста**
  - ▶ Меньше «перепасовок» между специалистами
  - ▶ Меньше специалистов требуется
  - ▶ К этому надо привыкнуть, сначала трудно
- **Практика важнее теории**
  - ▶ Существенное отличие от традиционного подхода
  - ▶ Для понимания Agile-подхода его надо увидеть в действии



# Причины успеха Agile-техник



# Текущие вызовы для Agile-подхода





# Agility зависит от динамичности проекта

## Организационные аспекты

Размер команды  
 Географическая распределенность  
 Организационная структура  
 Устойчивость существующих процессов

- Зрелые проекты
- 50+ разработчиков
- Сложные приложения
- Распределенная команда
- Масштабируемость, воспроизводимость, аудит

- Зрелые проекты
- Мультиплатформенные
- Сложные
- Распределенная команда
- Больше потребностей в координации

- Небольшая команда
- Новые проекты
- Простые приложения
- Расположены вместе
- Минимальные требования к документации

## Технические и правовые аспекты

Требования регуляторов  
 Управление  
 Сложность приложений

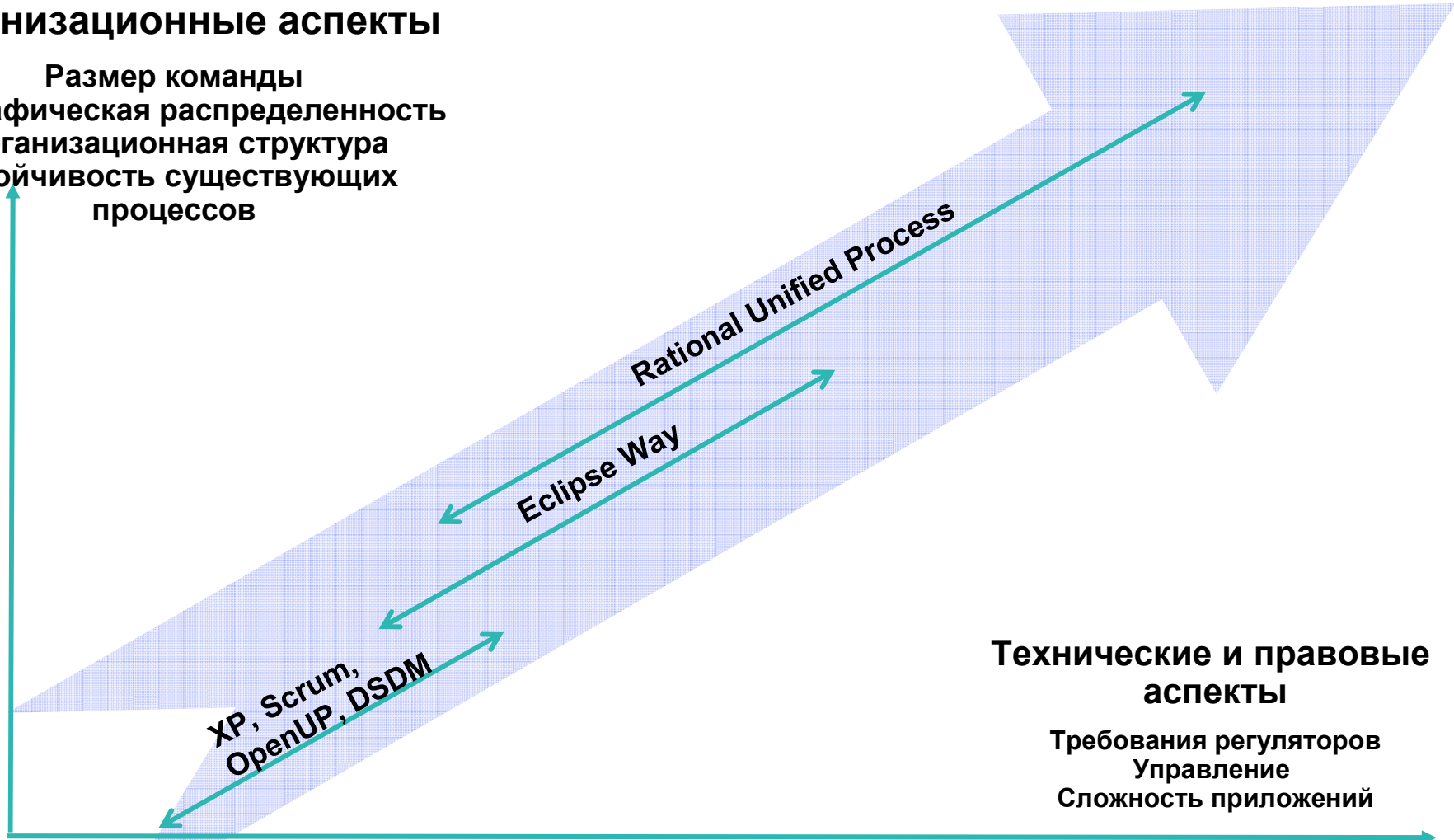




# Масштабируемость Agile-процессов

## Организационные аспекты

Размер команды  
Географическая распределенность  
Организационная структура  
Устойчивость существующих процессов



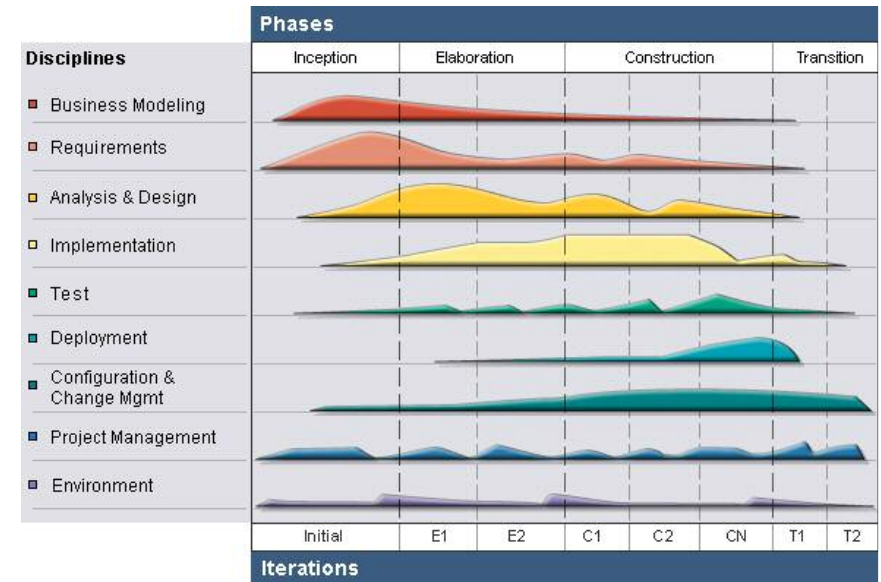
## Технические и правовые аспекты

Требования регуляторов  
Управление  
Сложность приложений

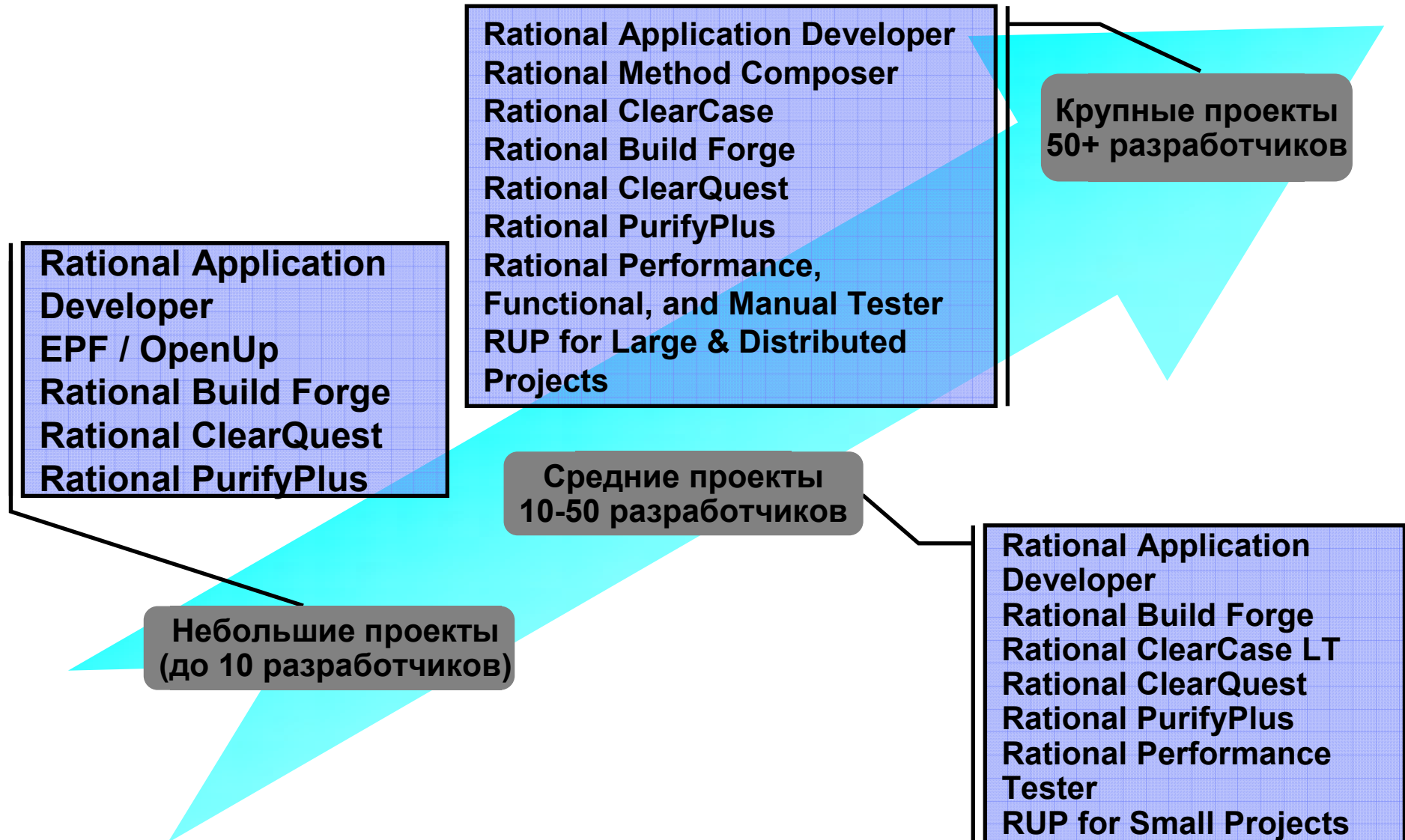


## Масштабирование Agile через Rational Unified Process

- RUP с успехом используется для agile-проектов
- Адаптация:
  - ▶ Регулярный выпуск работающего продукта
  - ▶ Активное участие представителей заказчика
  - ▶ Ежедневные обсуждения
  - ▶ Test-Driven Development (TDD)
  - ▶ Непрерывная сборка
  - ▶ Agile-документирование
- Преимущества масштабирования:
  - ▶ Бизнес-риски обрабатываются при инициации
  - ▶ Технические риски обрабатываются при проектировании



# Решения для Agile-проектов разной сложности



## Agile-процессы и RUP



**RUP for Large & Distributed Projects**



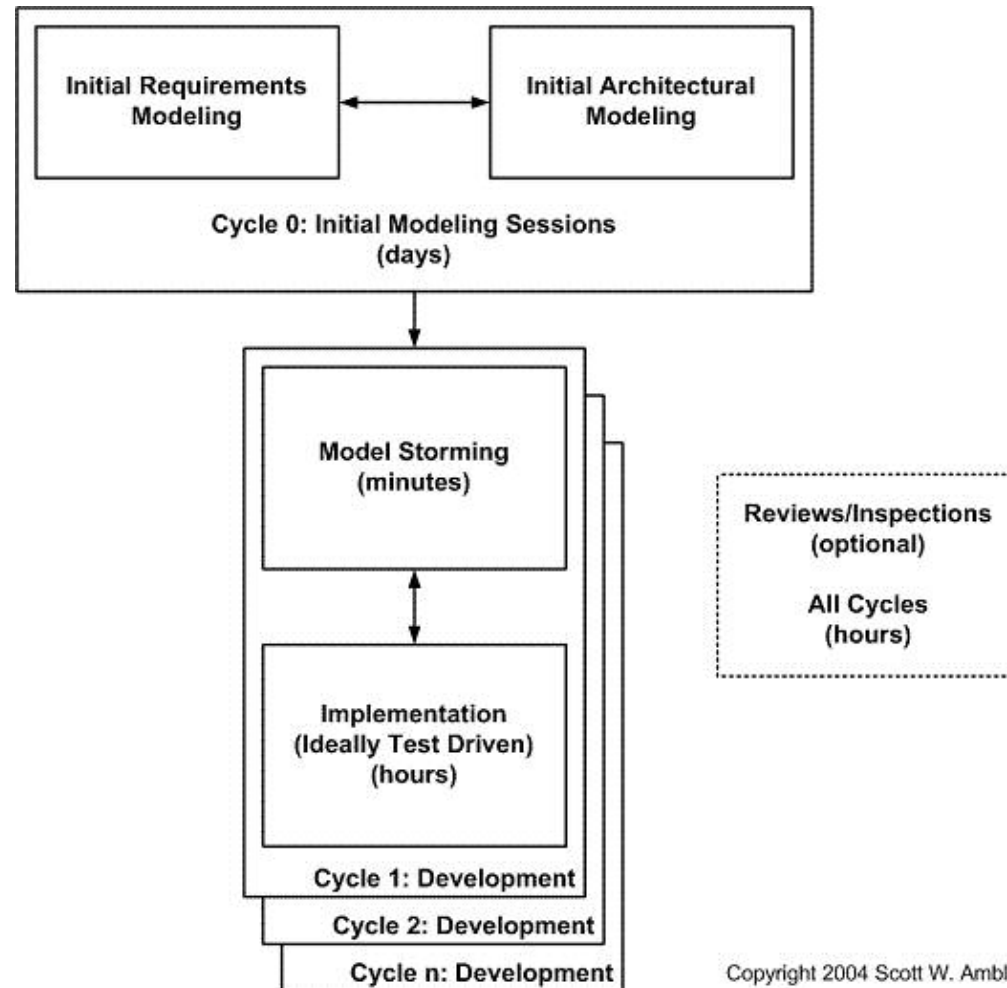
## Agile-документирование

- **Описание стабильных объектов**
- **Максимизация ROI для заказчика**
- **Лаконично**
- **Соответствует цели**
- **Описывает “что следует знать”**
- **Имеет определенного заказчика и облегчает работу этому заказчику**
- **В достаточной степени аккуратно, целостно и детально**
- **В достаточной степени индексируется**



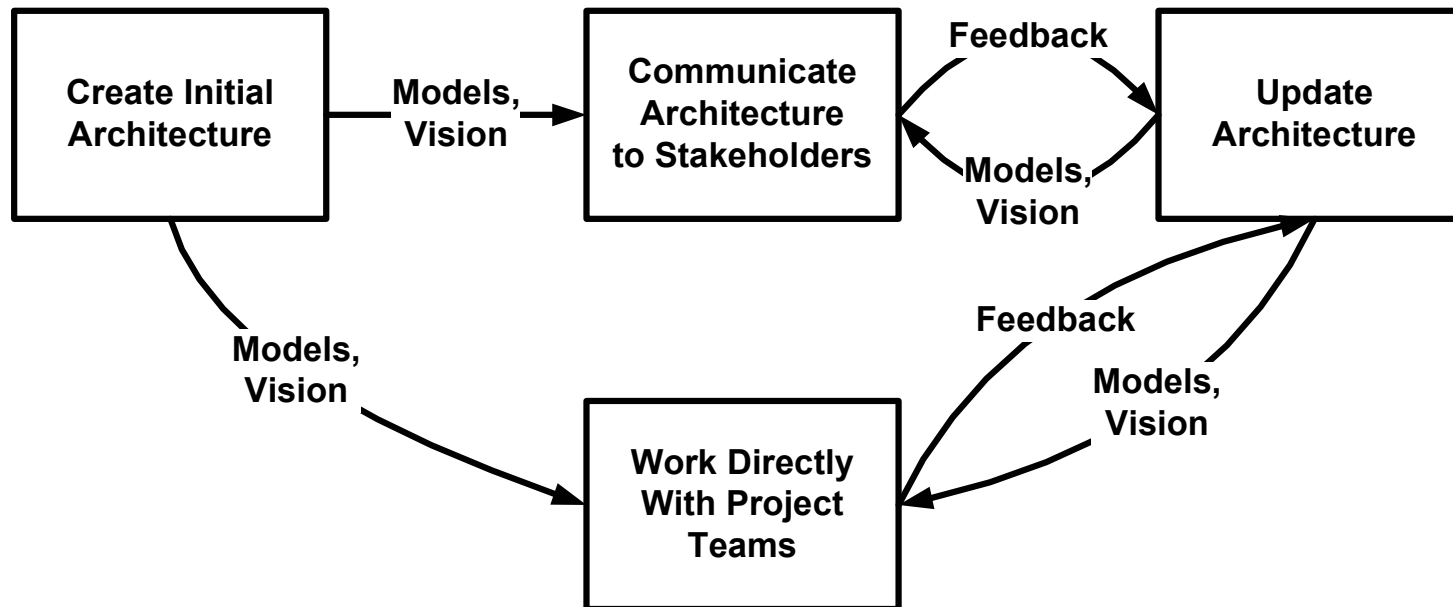
# Agile Model Driven Development (AMDD) – уровень проекта

[www.agilemodeling.com/essays/amdd.htm](http://www.agilemodeling.com/essays/amdd.htm)



# AMDD – уровень предприятия

[www.agiledata.org/essays/enterpriseArchitecture.html](http://www.agiledata.org/essays/enterpriseArchitecture.html)



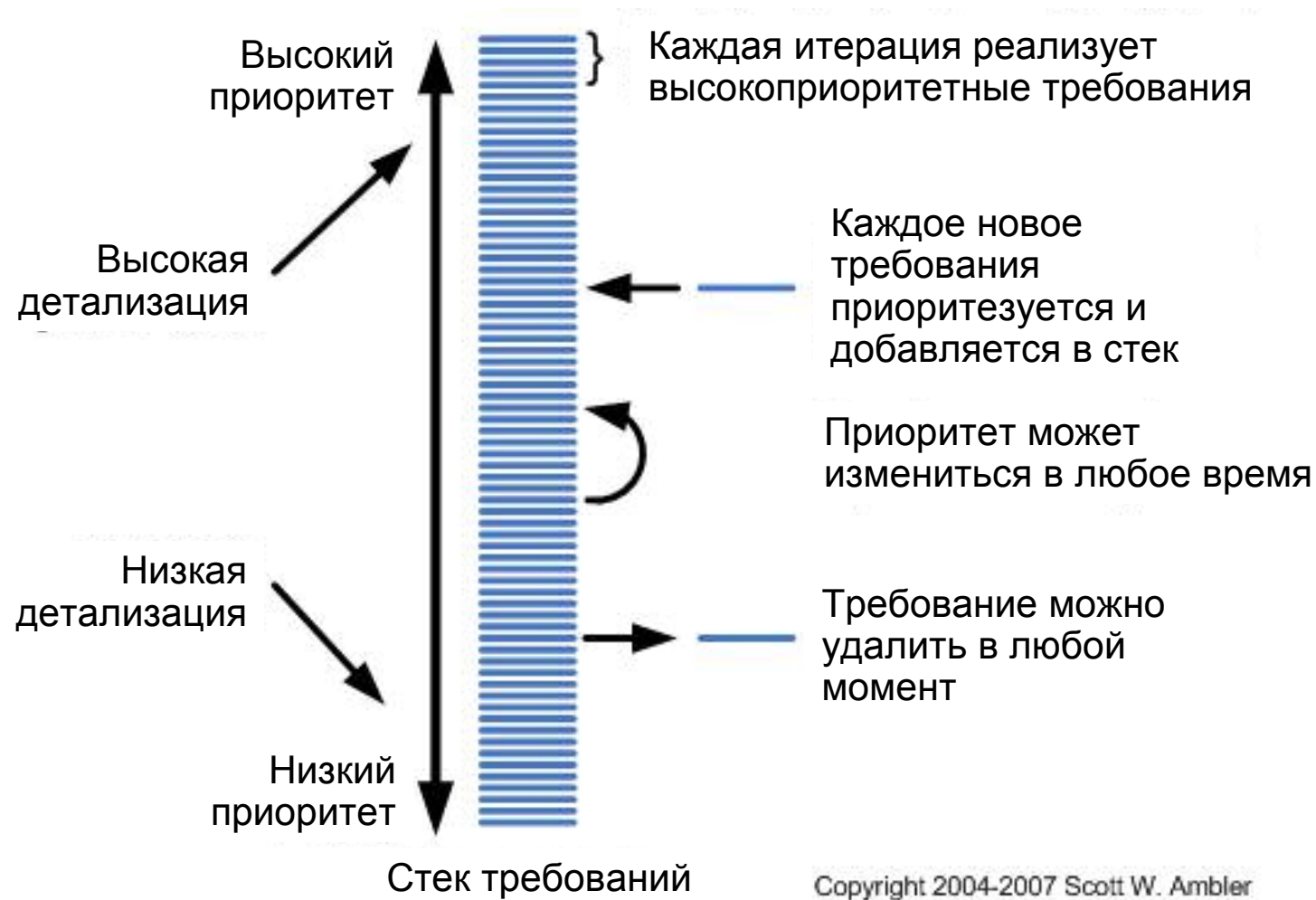
Enterprise architecture artifacts evolve over time





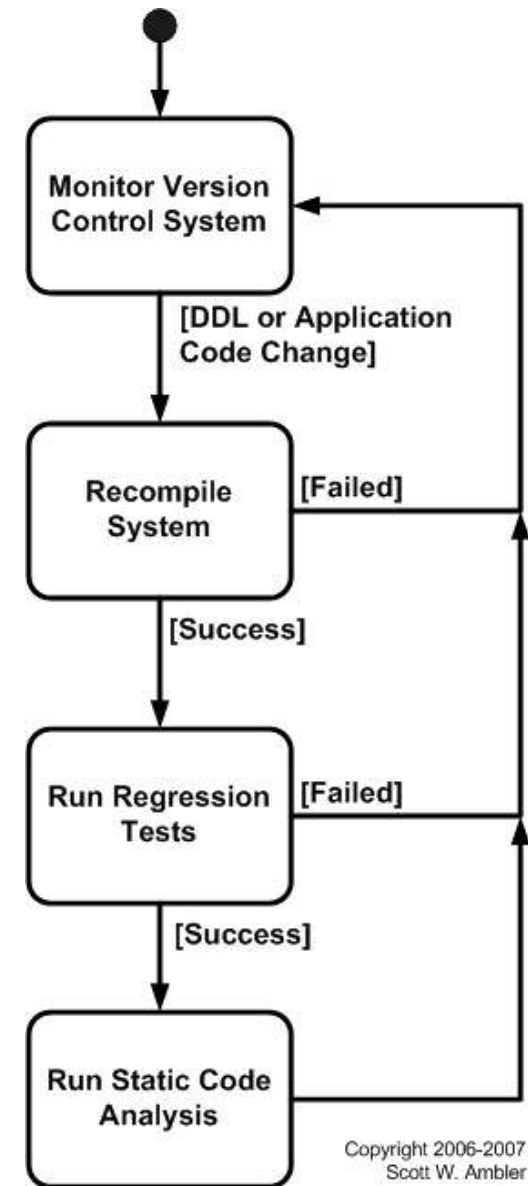
# Управление требованиями в Agile-проектах

[www.agilemodeling.com/essays/agileRequirements.htm](http://www.agilemodeling.com/essays/agileRequirements.htm)



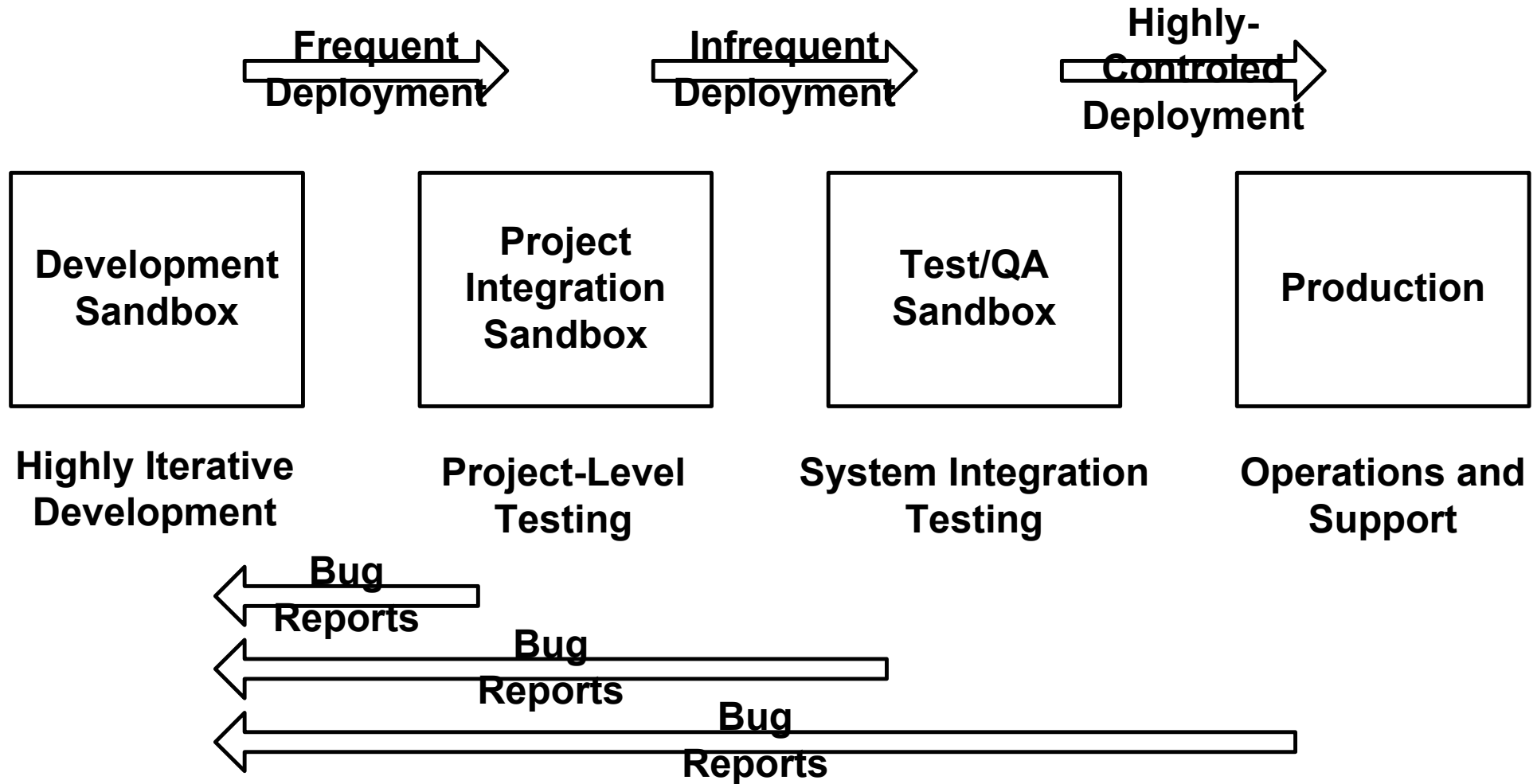
# Непрерывная интеграция

- При изменении исходных кодов система должна быть пересобрана
  - ▶ Часть такой пересборки является пересборка Базы Данных
- После пересборки - регрессионное тестирование
- Статический анализ кода – общепринятая практика для приложений
  - ▶ Тоже самое можно делать для баз данных
  - ▶ Тоже самое можно делать для интерфейса UI

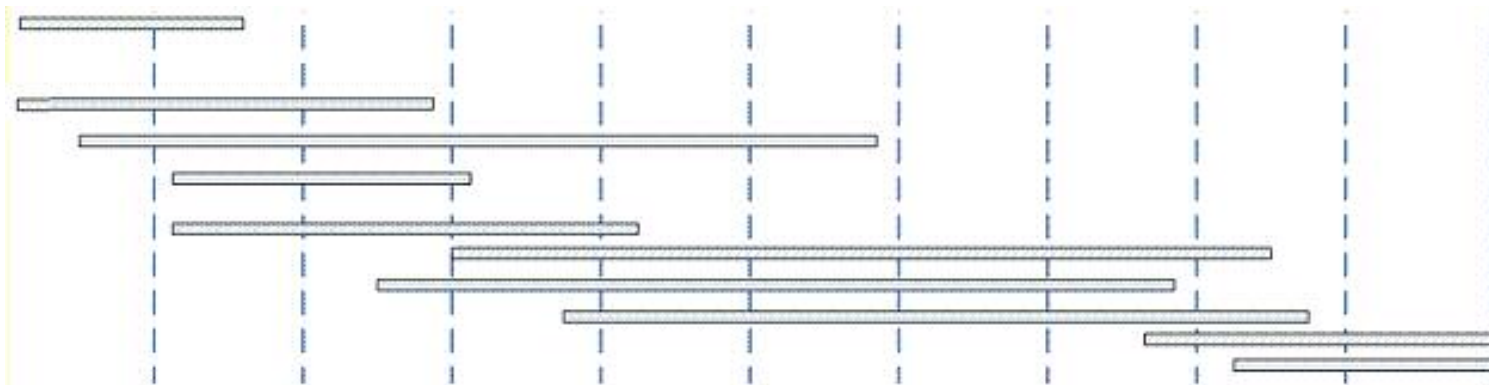
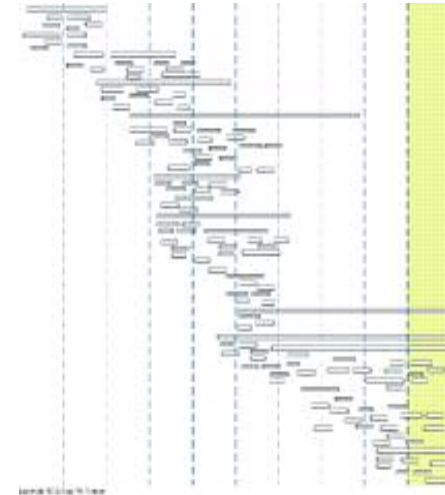
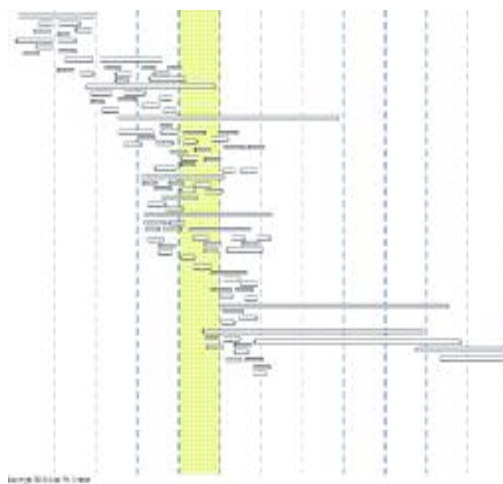
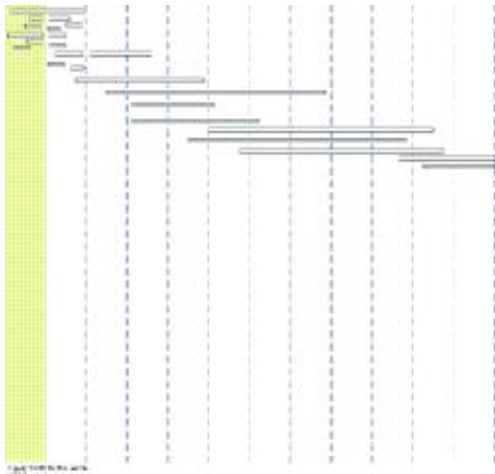


# «Песочницы» (Sandboxes)

[www.agiledata.org/essays/sandboxes.html](http://www.agiledata.org/essays/sandboxes.html)



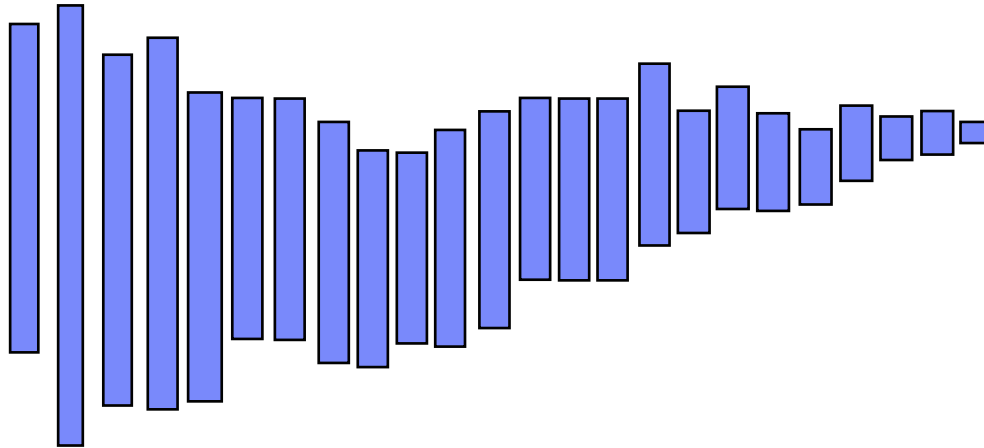
# Agile-подход в управлении проектами



Copyright 2005 Scott W. Ambler



# Оценка проекта

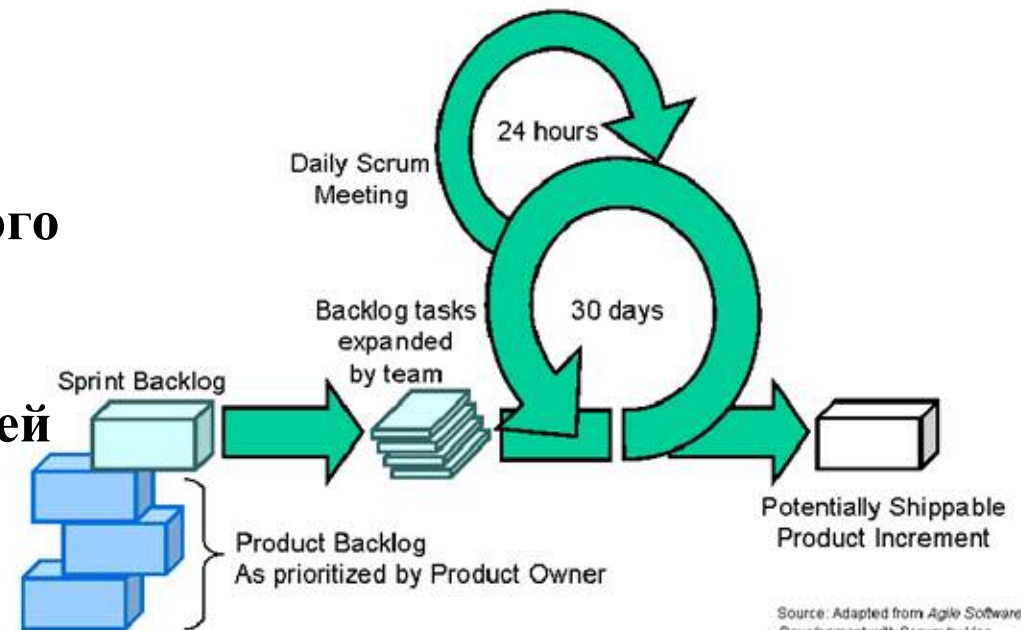
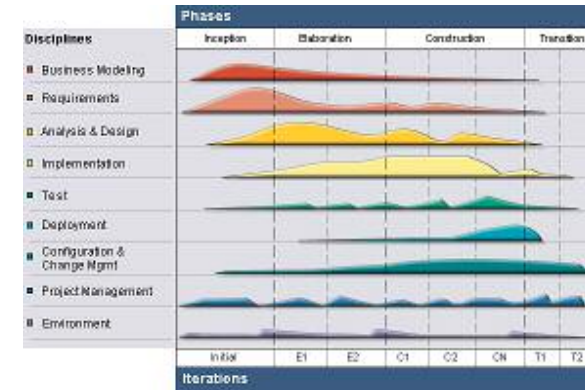


- Хорошая оценка выполняется:
  - ▶ Для небольших объектов
  - ▶ Людями, выполняющими эту работу
  - ▶ Тем, кто уже делал эту работу раньше
  - ▶ Just in time (JIT)



## Итерации

- Организуйте работу над проектом по “итерациям”
- Каждую итерацию заканчивайте выпуском работающего продукта
- Обеспечивайте управляемость:
  - ▶ Обратная связь для каждого участника
  - ▶ Актуальные данные по проекту для представителей заказчика
  - ▶ Контрольные точки



Source: Adapted from *Agile Software Development with Scrum* by Ken Schwaber and Mike Beedle.

THANK YOU

