

Ag;)eDays'10
Екатеринбург

Сбор и анализ требований в Scrum
Адаптация процесса ICONIX

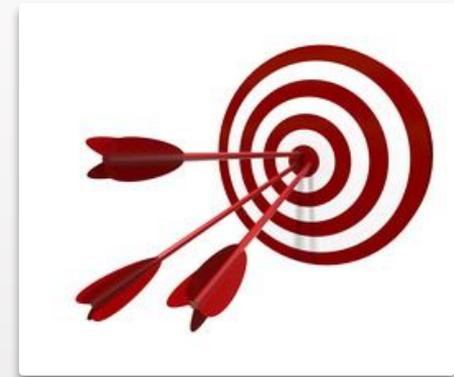
Вольфсон Борис

Руководитель проектов
Руководитель регионального отдела
веб-разработки

Компания Softline

Цели и содержания доклада

- Среда использования
- Описание процесса ICONIX
 - Диаграммы и процесс
- Адаптация процесса ICONIX под Scrum/Agile
 - Потери при производстве
 - Синхронизация диаграмм и кода
 - Соответствие принципам Agile
- Обсуждение и вопросы



Среда использования Scrum и ICONIX

- Компания Softline
- Разработка высоконагруженных коммерческих сайтов:
 - Корпоративные веб-сайты
 - Веб-сайты для электронной коммерции
- Около **100** основных участников проектов
 - Страны СНГ и дальнего зарубежья
- **Распределенная** команда разработки
 - Москва, Новосибирск и Оренбург

Сбор и анализ требований в Scrum

- Создание и нормализация видения продукта
- Выявление и описание персонажей
- Создание юзер-стори
 - Как «персонаж», я «действие» для «цель»
- Описания юзер-стори хранятся в виде «знаний» команды
 - Для распределенных команд удобно использовать вики



Сбор и анализ требований в Scrum

Плюсы

Никаких избыточных артефактов

Позднее принятие решений

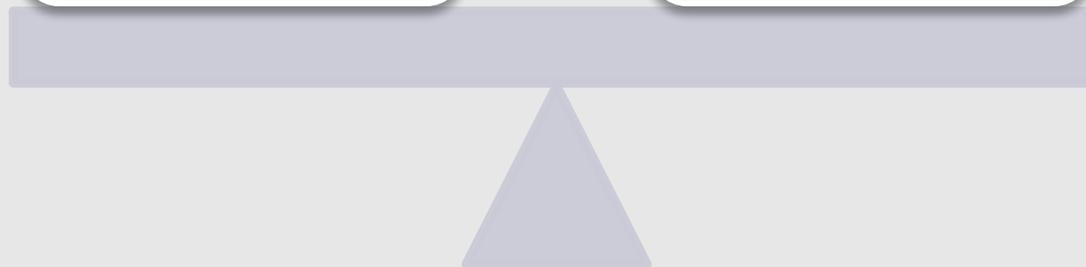
Быстрое создание и легкое управление

Минусы

Не очень подходит для распределенных команд

Отсутствует моделирование

Нет общего описания системы



Как мы понимаем Scrum

Бережливое
производство для
софтверных
проектов

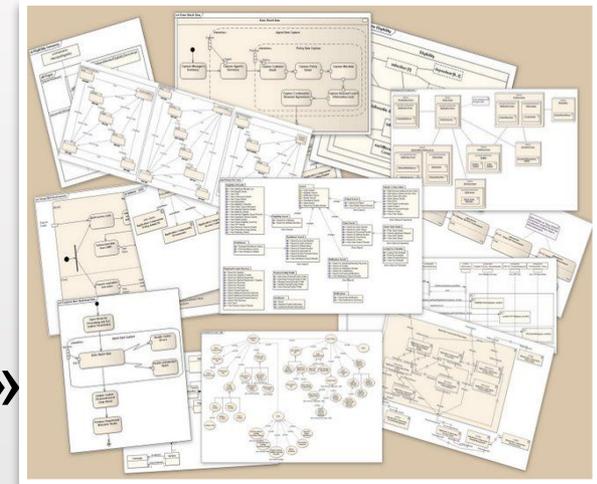
Вытягивающее
производство

Точно в срок (JIT)

перед
их
непосре
дственн
ым
использ
ованиём
••
Миним
альн
время
защасы
важные
••
итераци
и и
незав
исполь
заказчик
а –
потери,
которые
необход
ими
стран

Полная UML

- Большой входной порог
 - Более 10 видов диаграмм
 - 900-страничное руководство
- Слишком подробное описание
- Неявная «Водопадная модель»
- Избыточность
- Необходимость постоянной актуализации диаграмм



Потери при производстве: UML

- Перепроизводство
- Ожидание
- Переключение между задачами
- Лишние этапы обработки
- Лишние запасы
- Ненужные перемещения сотрудников
- Дефекты



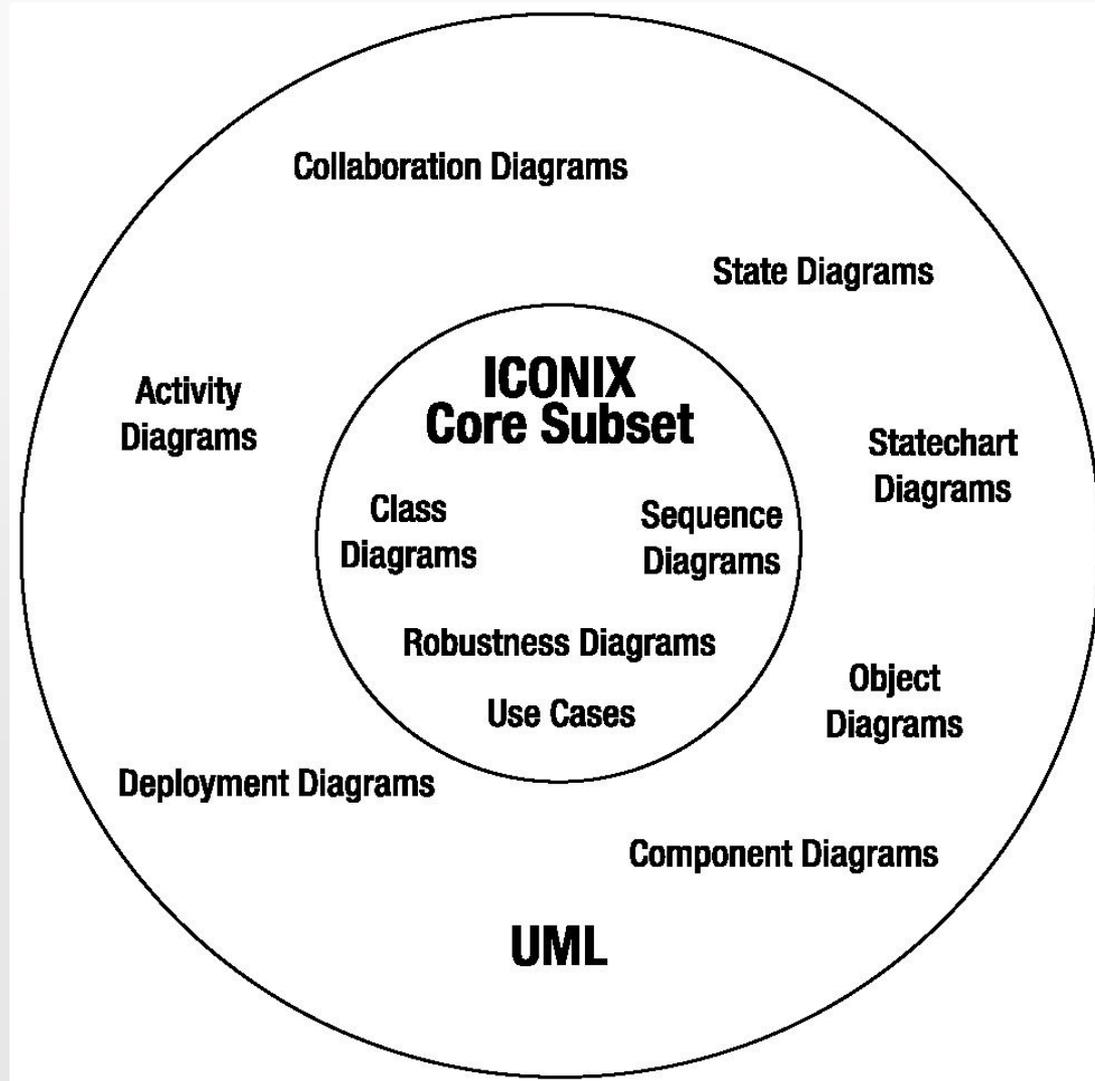
Что такое ICONIX?

Методология анализа требований,
основанная на вариантах использования

Описание процесса создания и
использования артефактов

Используется “подмножество” UML

ICONIX подмножество UML



Классическая схема процесса ICONIX

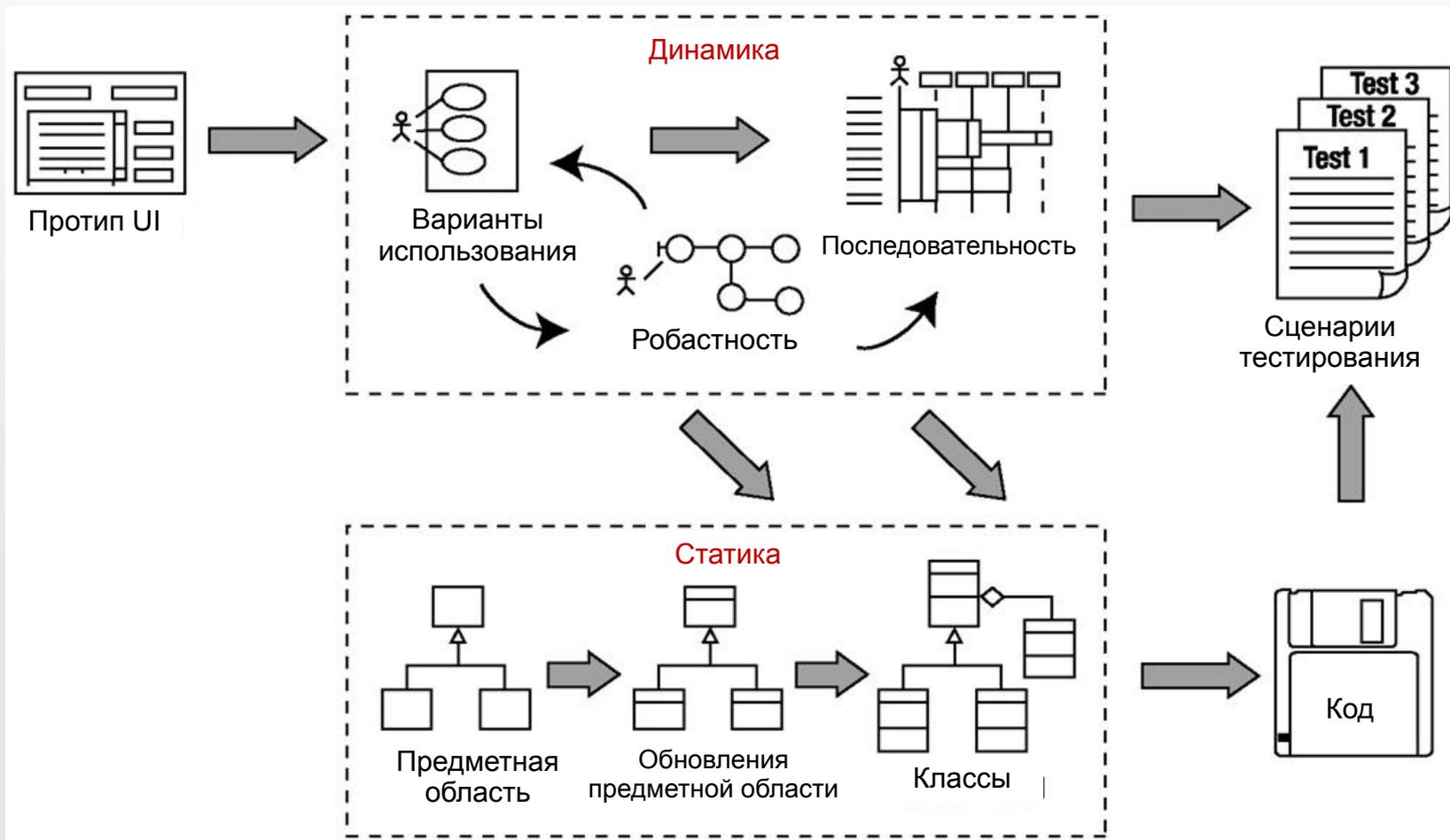


Диаграмма предметной области



Диаграмма классов

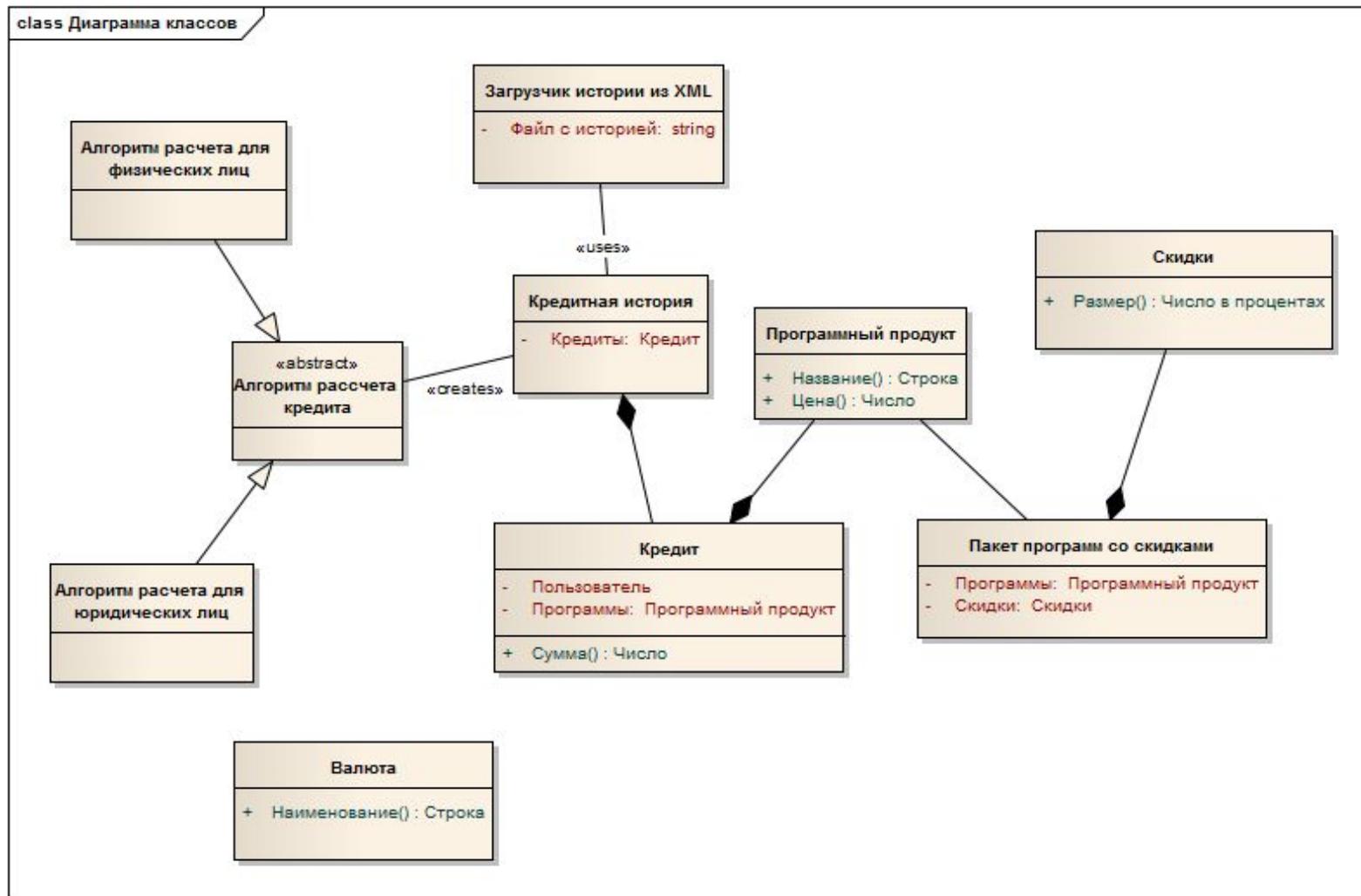


Диаграмма вариантов использования

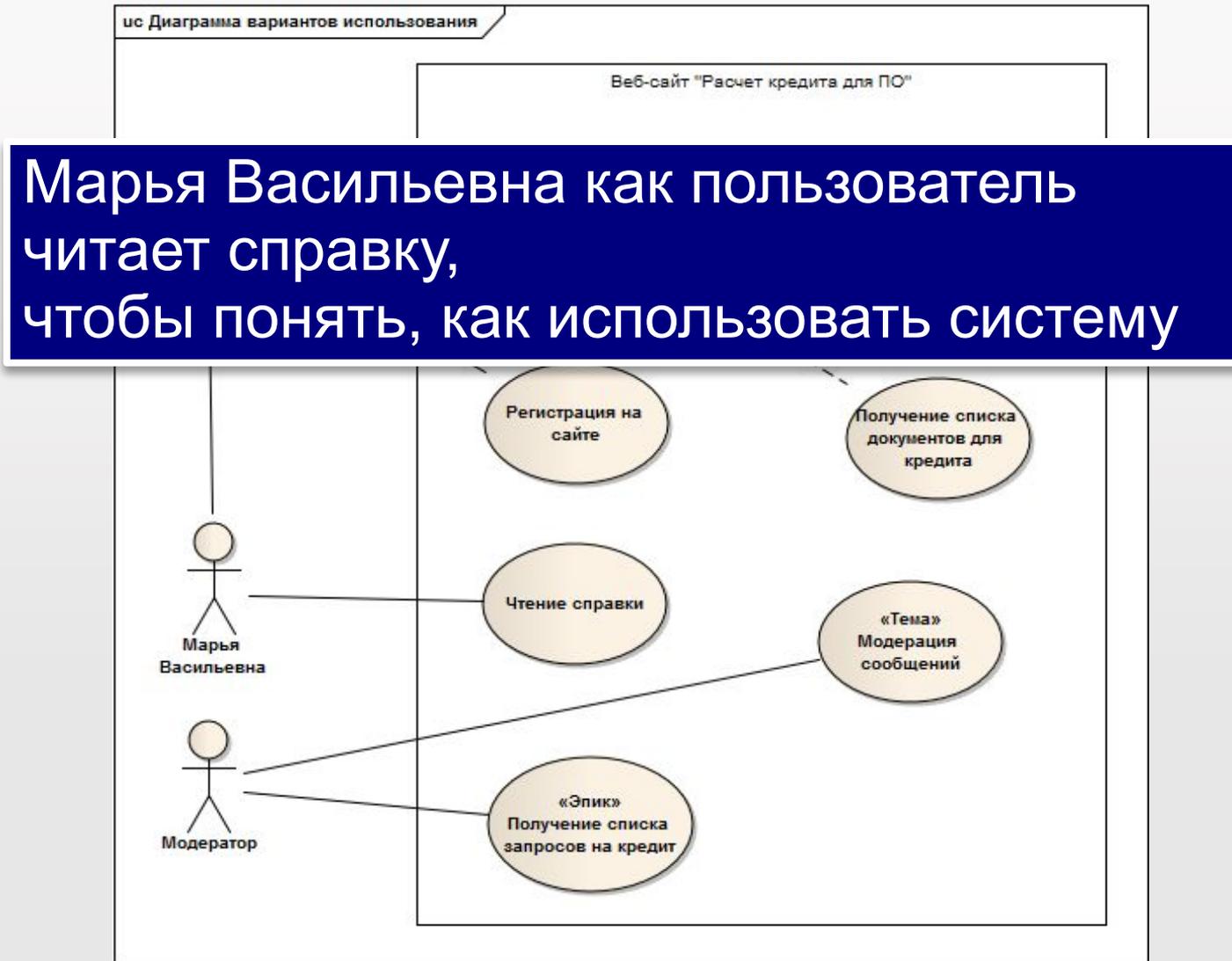
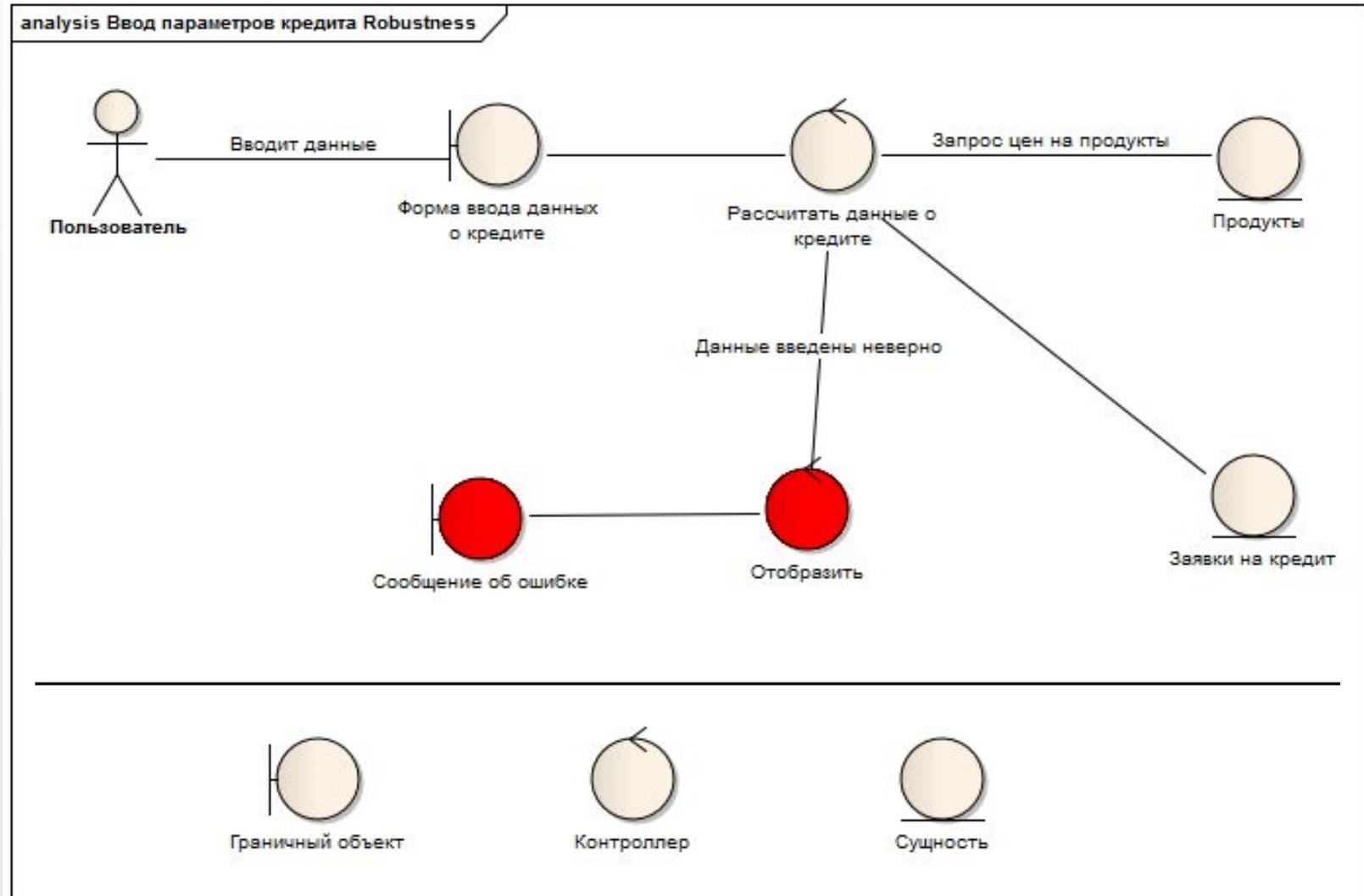


Диаграмма робастности



Зачем нужна диаграмма робастности?

- Проверка полноты юзкейсов
- Выявление дополнительных объектов
- Проверка текста юзкейсов
- Предварительная проработка архитектуры
 - «Мост» между анализом и архитектурой

Зачем нужна диаграмма робастности?

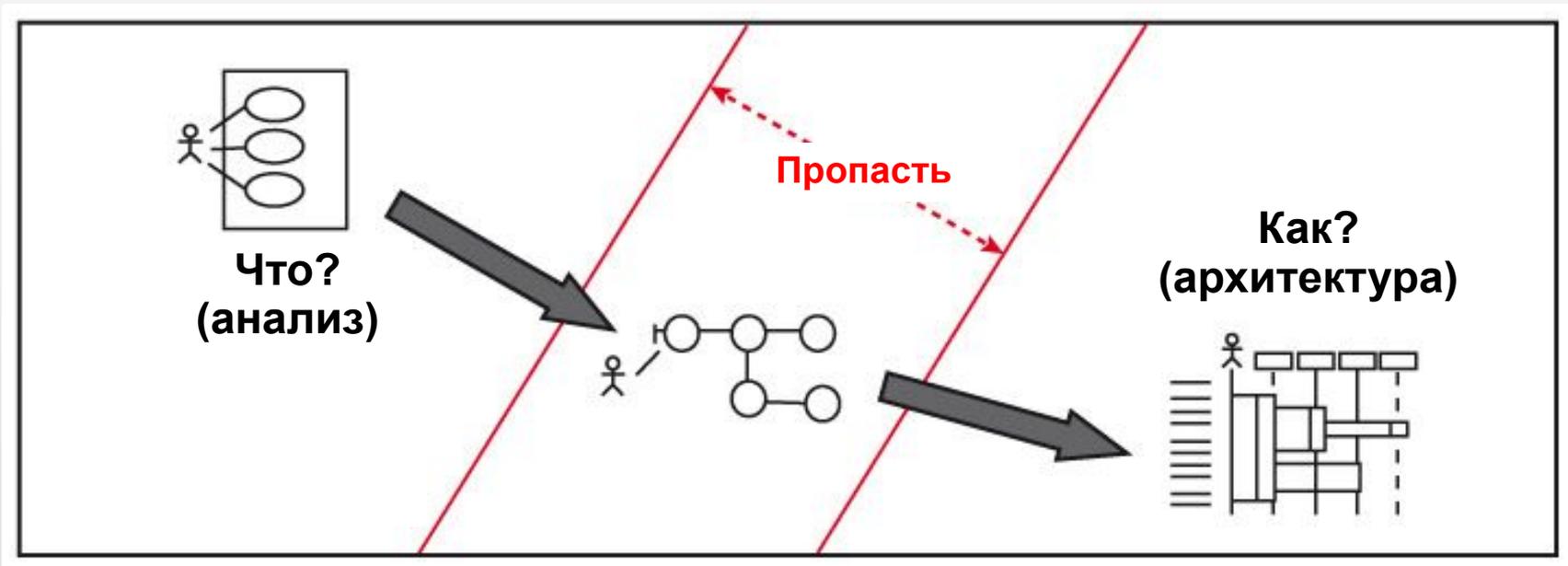
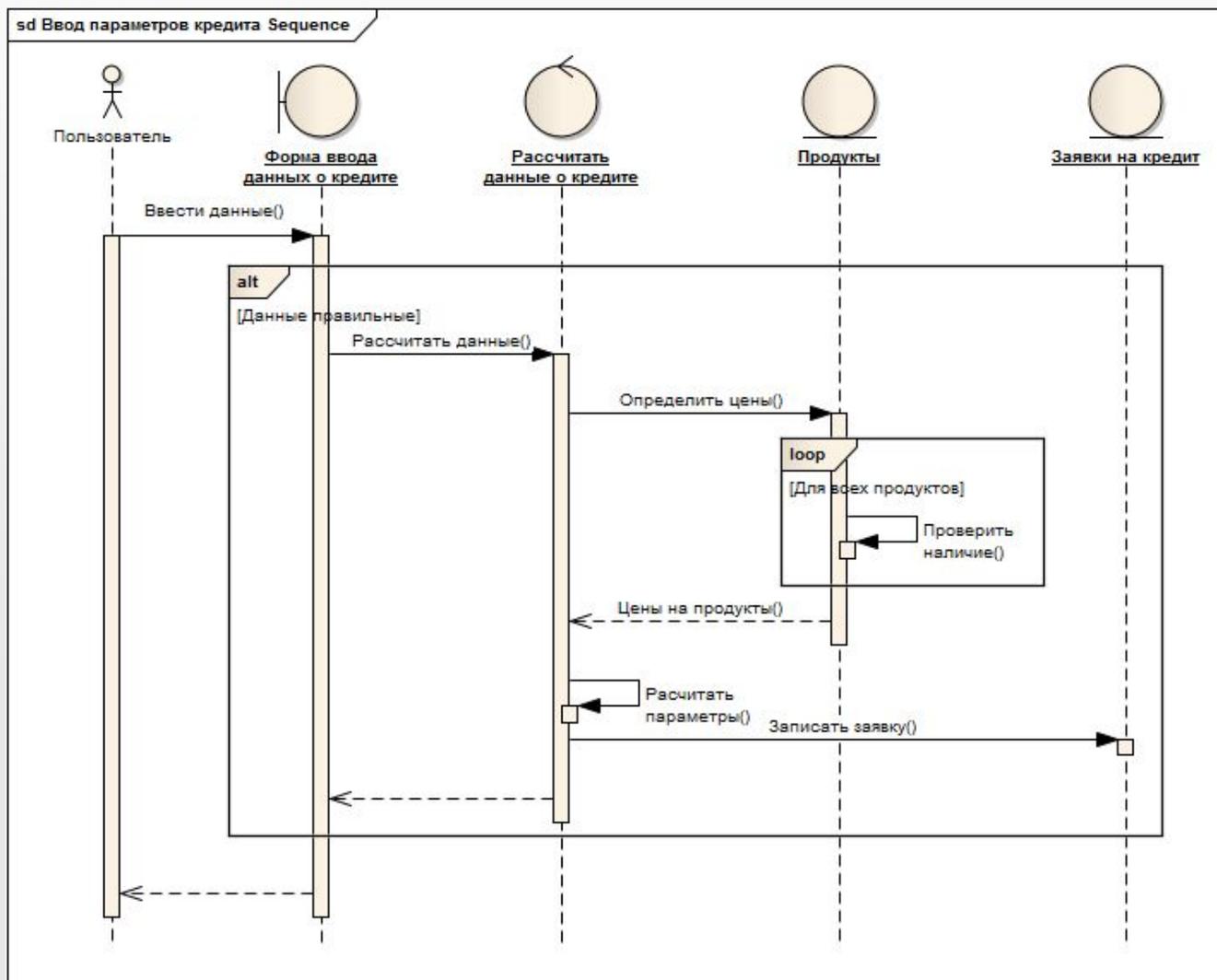


Диаграмма последовательности



Практики процесса ICONIX

- Анализ и уточнение требований
 - Системный аналитик для Product owner'а
 - Уменьшение количества неправильных требований
- Анализ предметной области
- Проектирование взаимодействия с системой
- Префакторинг – рефакторинг модели
- Синхронизация моделей и кода
- Агрессивное тестирование на всех уровнях

Проектирование взаимодействия с системой

До разработки

Во время разработки

Ранний фидбек

Понимание
продукта

UI на основе
требований

Вытягивающее
производство

Usability



Возвращение к водопадной модели?

Классический ICONIX:

- Близок к водопадной модели
- Допускает потери при производстве
 - Перепроизводство - проработка лишних требований
 - Лишняя обработка - актуализация диаграмм
 - Лишние запасы – проработка всей модели

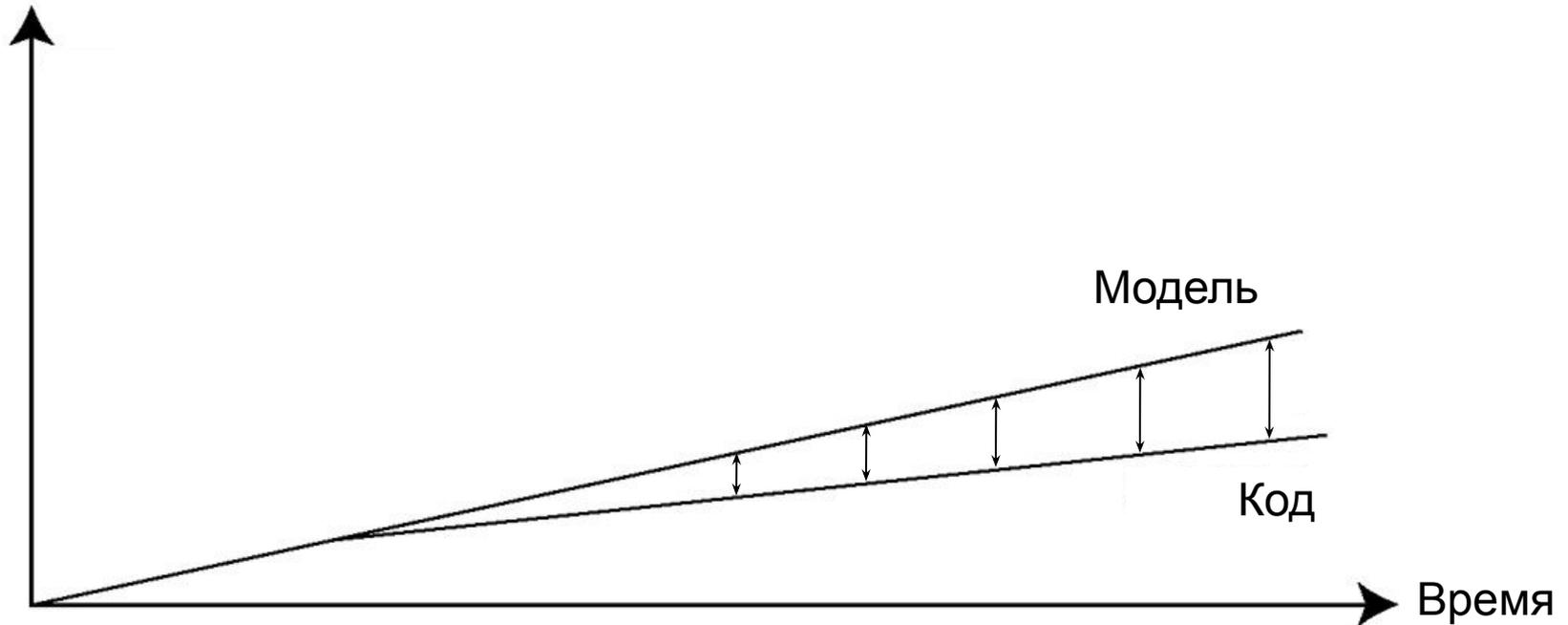
... но ICONIX отлично адаптируется к Agile

Варианты политик синхронизации диаграмм и кода

- Актуализация – **это потери!**
- Полная или частичная синхронизация
 - «Внешние разработчики»
 - Распределенная команда
 - Поддержка продукта
- Части продукта для синхронизации
 - Основной функционал
 - Взаимодействие с внешними системами

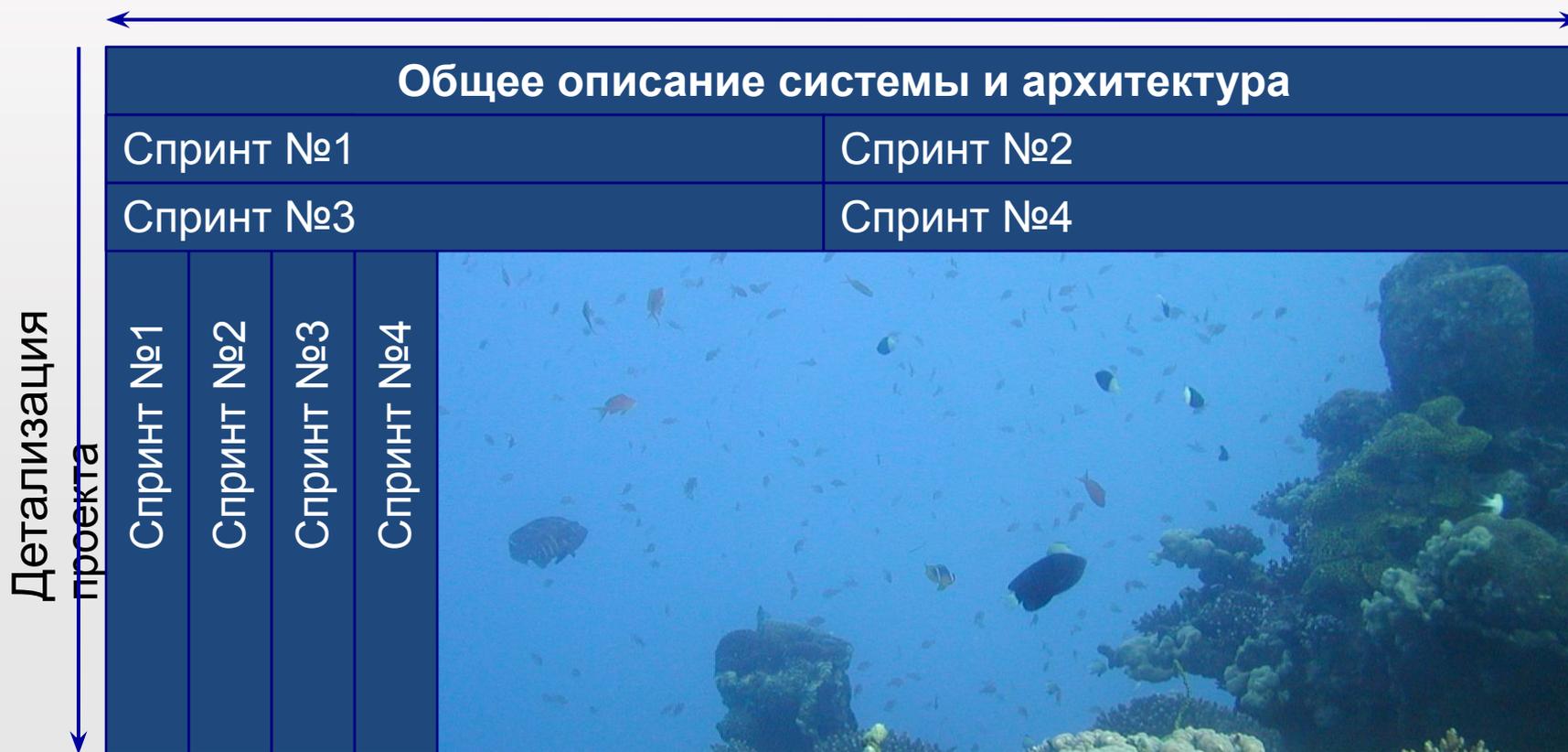
Различия между моделью и кодом

Количество
различий



Подводное плавание - метафора содержания проекта

Размер/рамки проекта



Нулевой спринт – плаваем на поверхности

- Видение продукта
- Диаграмма предметной области
- Диаграмма вариантов использования
 - Роли и персонажи
 - Юзер-стори без описания
- Проработка юзер-стори для первого спринта

Важно ограничить нулевой спринт **по времени**

Последующие спринты – ныряем на глубину

- Подробное описание юзер-стори
 - Не больше двух параграфов
 - Баланс текстового и графического описания
- Диаграмма робастности
- Диаграмма последовательности
- Диаграмма классов
- Обновление диаграммы предметной области и диаграммы юзкейсов



Возможные опасности

Неполная осведомленность МП о возможностях
продукта

Ослабление коммуникаций между МП и командой

Преждевременная проработка лишних
требований

Паралич анализа на нулевой итерации

Agile Manifesto

- **Люди и их взаимодействие**
важнее процессов и инструментов
- **Готовый продукт**
важнее полной документации
- **Сотрудничество с заказчиком**
важнее контрактных ограничений
- **Реакция на изменения**
важнее следования плану



Инструменты



Бумага и карандаш



Visio



Enterprise Architect



Плюсы и минусы ICONIX

Плюсы

Минусы

Моделирование в виде
наглядных диаграмм

Подмножество
стандартного языка UML

Сочетаемость с Agile-
методологиями

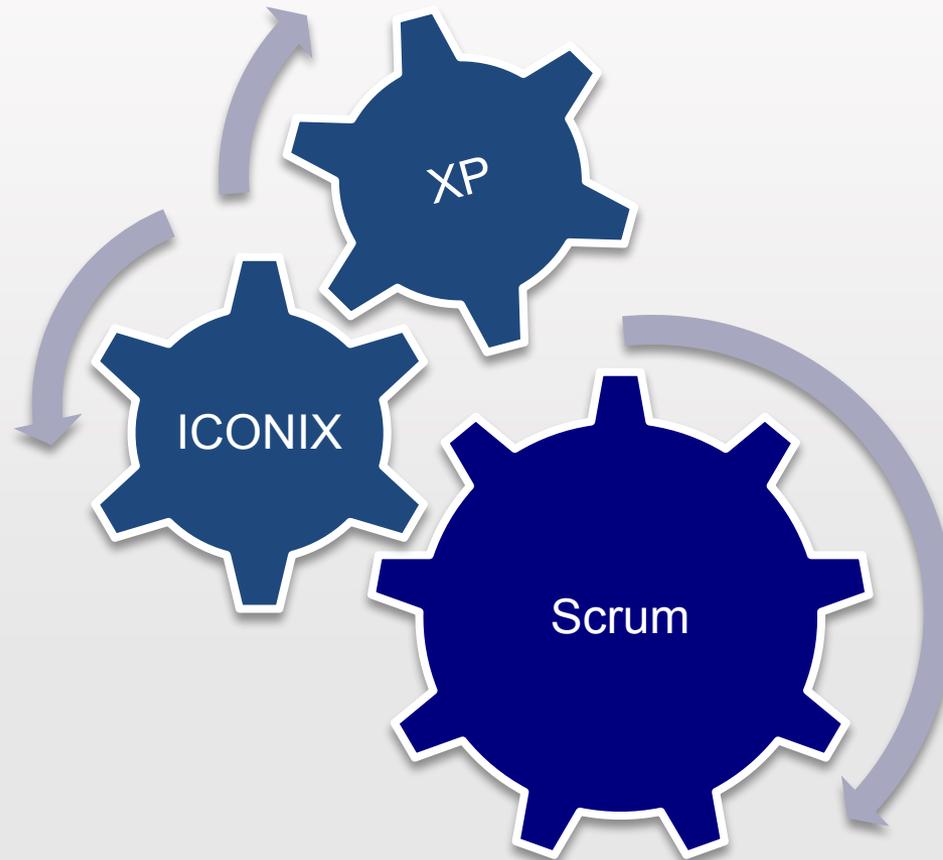
Подходит для
распределенных команд

Необходимость
актуализации моделей

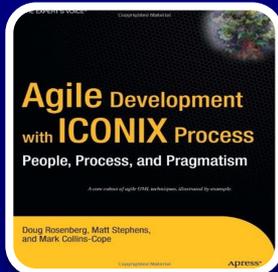
Диаграммы не нужны
заказчику

Нарушение абсолютной
кроссфункциональности
команды

Методологии

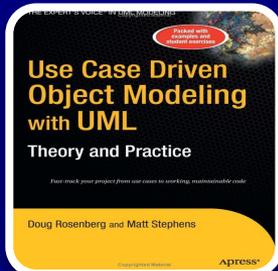


Литература



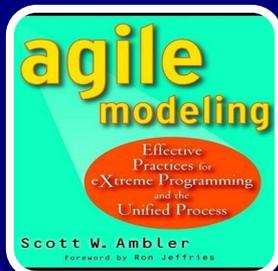
Agile Development with ICONIX Process: People, Process, and Pragmatism

by Doug Rosenberg, Matt Stephens and Mark Collins-Cope



Use Case Driven Object Modeling with UML: Theory and Practice

by Doug Rosenberg and Matt Stephens



Agile Modeling: Effective Practices for eXtreme Programming and the Unified Process

Scott W. Ambler

Контакты и вопросы

- Спасибо за внимание!
- Вопросы?

Мои контакты

borisvolfson@gmail.com

borisv@softline.ru

www.twitter.com/_blv_

Ag;)eDays Екатеринбург'10

