

Одиннадцатая независимая научно-практическая конференция «Разработка ПО 2015»

22 - 24 октября, Москва



# Внедрение agile в большой корпорации: приключения обыкновенные и невероятные

Оксана Некрасова, Иван Гуменюк

EMC

# Размер имеет значение

- Огромное количество разработчиков (~1000)
- В некоторых областях требуется экспертиза, которая наработывается несколько лет
- Огромная кодовая база (миллионы строк кода), часть кода существует уже много лет
- Архитектурный беспорядок, часть кода унаследована с «античных» времен
- Большое количество различных технологий

# Сложности

Релизы выходят редко и с большим опозданием

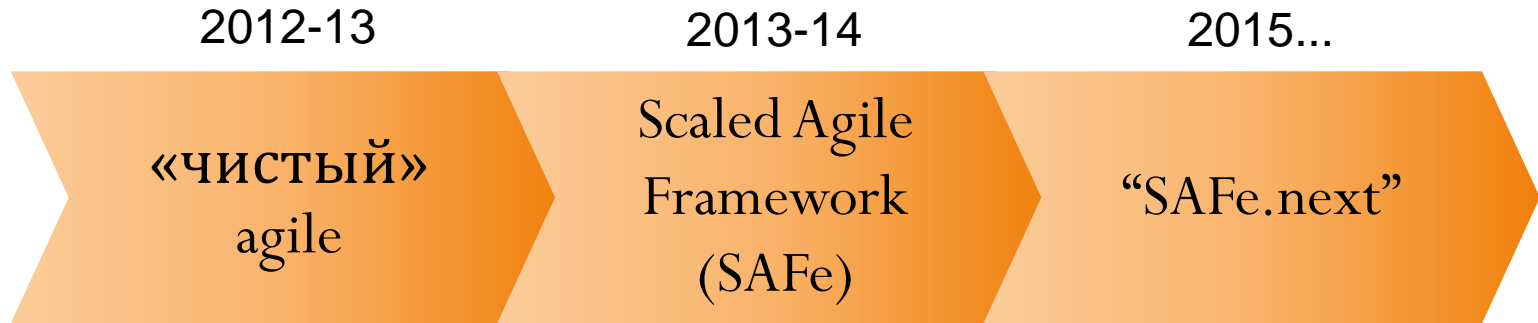
Результат не соответствует ожиданиям Product Managers

Негибкость к изменяющимся требованиям

Слабое внедрение инноваций



# Шаги 'перестройки'



*«Приехавший сказал: О том, что я видел, я мог бы говорить целый год. Суть в том, что никто кроме нас не знал где здесь выход и даже мы не знали где вход.»*  
(Б.Г.)

# Первая этап: «чистый» agile

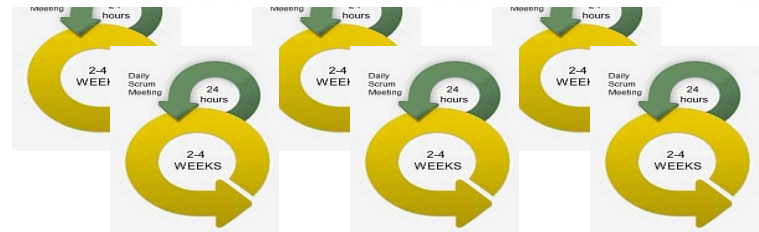


Никогда не сдавайся

# Приключение 1: первый блин комом



- Организовали scrum команды
- Зафиксировали цикл разработки
- Зафиксировали все основные практики: daily scrum, planning, retrospective, demo
- Установили цели разработки
- Запустили



# Анализ результатов

- Agile в «чистом» виде подходит для небольших организаций
- В корпорации необходима четкая синхронизации команд и правильная орг.структура
- Коллективная ответственность - должна распространяться не только на каждую команду, но и на всю работу целиком, чтобы избежать «finger pointers»
- Единый набор приоритетов для всей организации



## Второй этап: SAFe



Нельзя сдаваться не только  
после одного, но и после ста поражений.



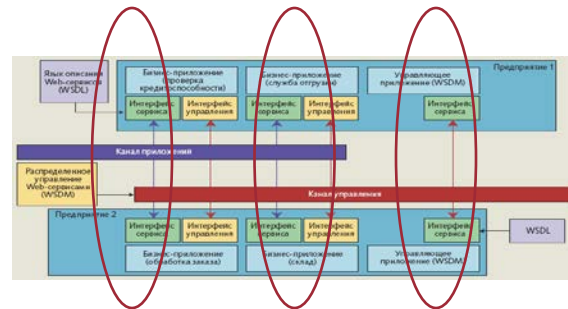


# Приключение 2: построение орг.структуры

## Модульная разработка (architecture):

Традиционно команды разбиваются по архитектурным модулям. Каждая команда отвечает за один модуль.

Интеграция моделей – отдельная непростая задача.



## End-to-End разработка (business value):

Функциональность разрабатывается целиком через все модули, упрощая получение value. Этап интеграции упрощается. Но, в одной команде надо объединить людей со знаниями из разных областей...

# Проблемы End-to-End подхода



## Технические

- Теряется архитектурная целостность
- Переработка framework затрагивает все команды
- Накопление технического долга

## Человеческие

- Требуются знания из многих областей
- Тестирование также ответственность команды разработчиков
- Команда перегружена работами по поддержке business value

# Приключение 3: Масштабирование процессов



## Agile manifesto:

- Работающий продукт важнее исчерпывающей документации
- Сотрудничество с заказчиком важнее согласования условий контракта

## Время на полноценное demo:

100 команд x 30 минут = 50 часов!!!

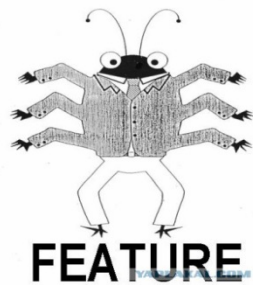
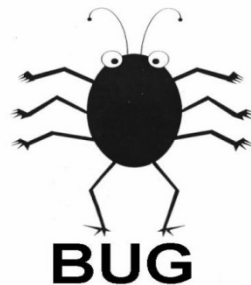
(с участием заказчиков, менеджеров, представителей других команд)





## Приключение 4: Качество продукта

- Децентрализация принятия решений
- Команда отвечает за business value и качество



Что такое – работающий продукт?

Каждая команда имеет свое представление о качестве, архитектуре и подходам к тестированию

=> Потеря единого представления о качестве системы



# Тестирование: больше вопросов, чем ответов

Как определить границы ответственности команды?

- Как долго команда сопровождает код?
- Кто отвечает за cross-functional тестирование?
- Есть ли команда системного тестирования и за что она отвечает?

Что такое «функциональность готова» с учетом configuration management?

- На каком branch/stream функциональность считается готовой?
- Слияние двух фич и возникающие при этом проблемы

Как тестировать зависимости (алгоритм приемки)?

# Приключение 5: Люди и документация



**Agile manifesto:** Люди и взаимодействие важнее процессов и инструментов

## Процессная

- Как объяснить командам (1000 человек) основные правила (DoD, необходимые действия в конце sprint, и т.д.)?
- Как согласовать изменения процесса?

## Техническая

- Как взаимодействовать с заказчиками на удаленных сайтах?
- Как запомнить нюансы 30 фич, идущих в релиз?

# Приключение 6: Overhead активности

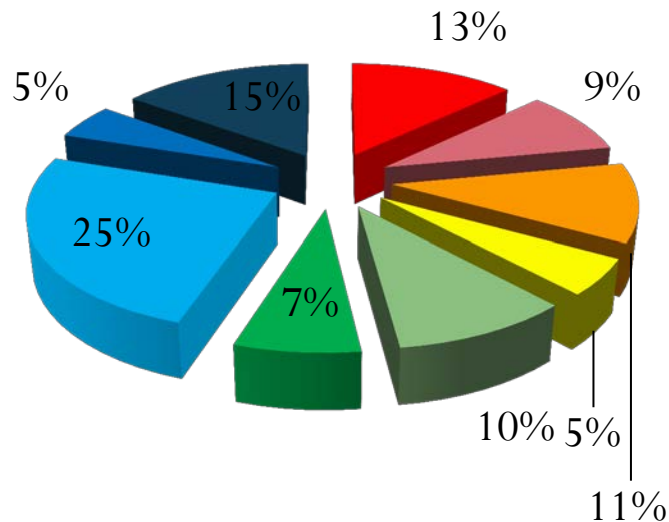


## Ожидания:

- Agile церемонии
- Тестирование

## Реальность:

- Интеграция между командами
- Зависимости и их согласование
- Заккрытие технического долга
- SME активности
- Регрессионное тестирование
- Сопровождение кода





# Преимущества agile

- Мотивированность команд и их ориентация на результат
- Прозрачность всех процессов
- База исторических данных
- Прогнозирование на основании предыдущего опыта
- Изменения легко контролируются, используя факт данные



Ни одна практика не является догмой. Фактические данные помогают проверить ее эффективность







# Приключение 7: Непрерывность изменения

## Кто изменяет

- Команда предлагает
- Менеджеры принимают решение:
  - Используя наглядность текущих процессов
  - Разбираясь в причинах запроса

## Причины запроса на изменения

- Практика понятия неправильно
- Практика не работает в данных условиях
- **Именно здесь кроется причина торможения команды**

# Третий этап: continuous improvement



[www.koubstudio.ru](http://www.koubstudio.ru)

**Koub** Studio  
motivators.ru

**ВСЕГДА** есть способ достигнуть  
цели... наиболее оптимальным путем!!!



# Гибридный подход к построению команд



- В гибридной модели есть функциональные команды и архитектурные (framework)
- Специализацию в команде не устранить, планирование при этом усложняется, т.к. количество работы в разных областях – разное
- Velocity индивидуальное
- Пересборка команд в зависимости от задач



# Документация



## Процессная:

- процессная книга – справочник
- процессная книга обновляется каждые три месяца перед очередным планированием

## Техническая:

- Функциональная спецификация
- Обновляется одновременно с написанием кода и тестов

# Адаптация SAFe



## Орг. структура:

- Добавлен еще один уровень объединения команд по функциональным областям (домен). Определены РО, RTE, архитектор для каждого домена

## Планирование:

- Проводится независимо сначала в рамках команды, затем в рамках функциональной области и только после этого в рамках организации
- Единый список приоритетов!
- Этапы планирования разнесены по времени



# Адаптация SAFe (продолжение)

## Демо:

- Проводится на уровне функциональных областей

## Тестирование:

- Каждой функциональной области в дополнение к архитектору добавлен тестовый архитектор
- Распределены роли между scrum командами и организацией системного тестирования
- Добавлен RPO – Release Product Owner (RTE управляют доменами)



# Приключение 8: Совмещение процессов

## Traditional Waterfall Methodology



- Sales используют водопадную модель (релизы привязаны к датам trade shows и marketing events)
- Разработчики используют свою модель, которую надо вписать в «водопад»

# Временная диаграмма

