



#SECONRU



МЕЖРЕГИОНАЛЬНАЯ КОНФЕРЕНЦИЯ  
РАЗРАБОТЧИКОВ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

# Как добавить работе по Agile предсказуемости, не потеряв гибкость

Цветчих Денис

Руководитель проектов в компании OneSystems

ПЕНЗА



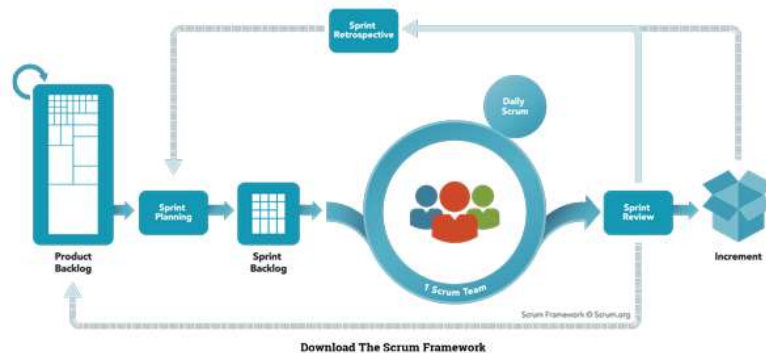
# Кто я

- Меня зовут Денис и я руковожу проектами уже 4 года
- Опыт как в аутсорсинге, так и в продуктовой разработке
- Опыт работы и по тяжелым методологиям, и по Agile



# В нашей компании некоторое время назад

- 1 заказчик
- 3 команды
- Scrum по евангелию
- Один филиал в Новосибирском академе
- ПМ вообще не было



# Со временем

- Нас стало много (более 15 команд)
- Стало много заказчиков
- Появились ПМ
- Филиалы Челябинске, Питере и Новосибирске в городе
- Специализированные команды с отдельными бэклогами
- Команды не кроссфункциональные (QA отдельно от Dev)



# Особенности

Кровавый Enterprise (рынок страхования)

Мы разрабатываем продукт

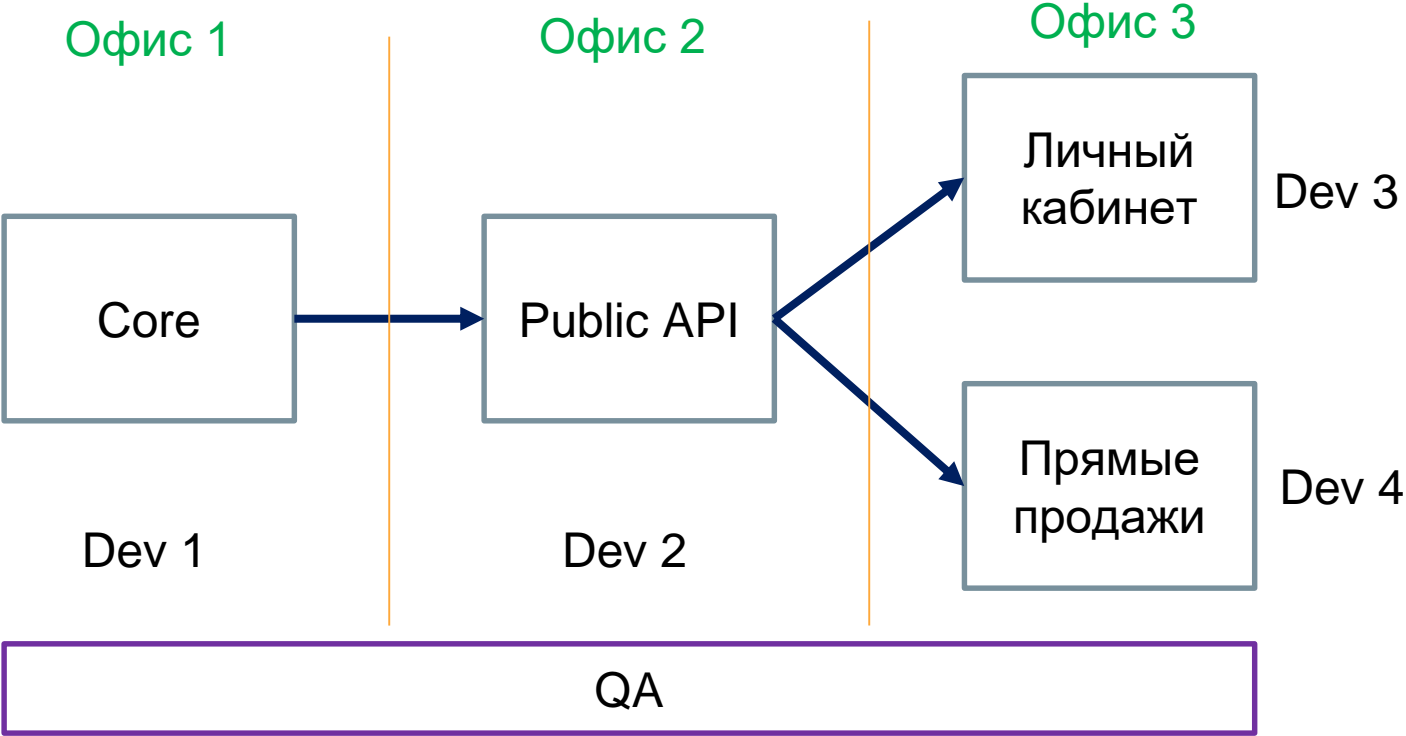
Но продукт дорабатывается для каждого кастомера

Перед кастомерами есть обязательства по задачам и срокам

Ежемесячные релизы



# Специализированные команды



# МЫ ХОТИМ

- Сократить время выхода задач, затрагивающих все проекты
- Водопад с четким ТЗ – однозначно сделать не то
- Писать ТЗ на 100 страниц – не попасть ни в какие сроки
- Укладываться в дедлайны для задач

Нам нужно масштабирование Agile!





# О чем поговорим

Каким бывает масштабирование Agile

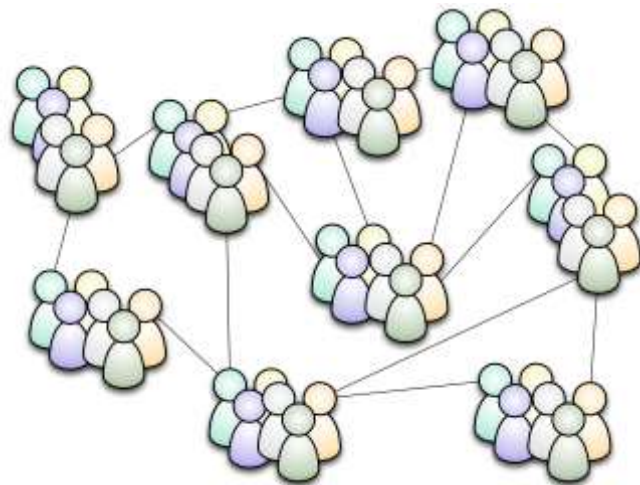
- Scrum of Scrums
- Nexus
- LeSS
- SAFe

С чего начать и какие будут трудности



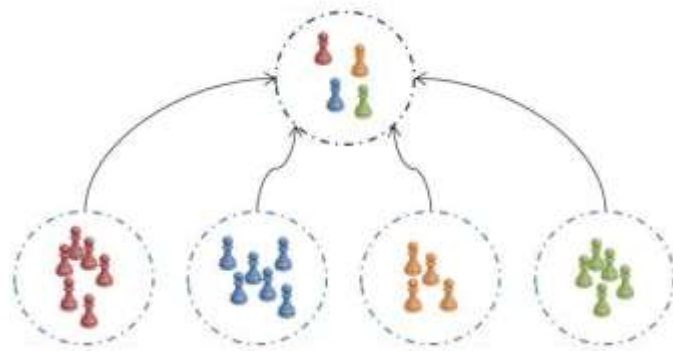
# Опрос

- Кто ничего не знает о масштабировании Agile?
- Кто слышал, но не применял?
- У кого есть опыт применения?



# Scrum of Scrums

Митинг скрам-мастеров



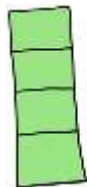
Как часто проводить, не регламентируется (как правило каждый день)

Что обсуждать – тоже не регламентируется, но как правило:

- Что сделали с прошлого собрания
- Какие проблемы (в том числе во взаимодействии команд)
- Что сделаем к следующему собранию

# Это круто, когда:

- Общий бэклог на все команды



- Команды взаимозаменяемы



# Не описывается

Одновременный ли старт спринтов в разных командах?

Как часто проводить Scrum of Scrums?

Каким составом его проводить?

Какие вопросы обсуждать?

Кроссфункциональные команды или нет?

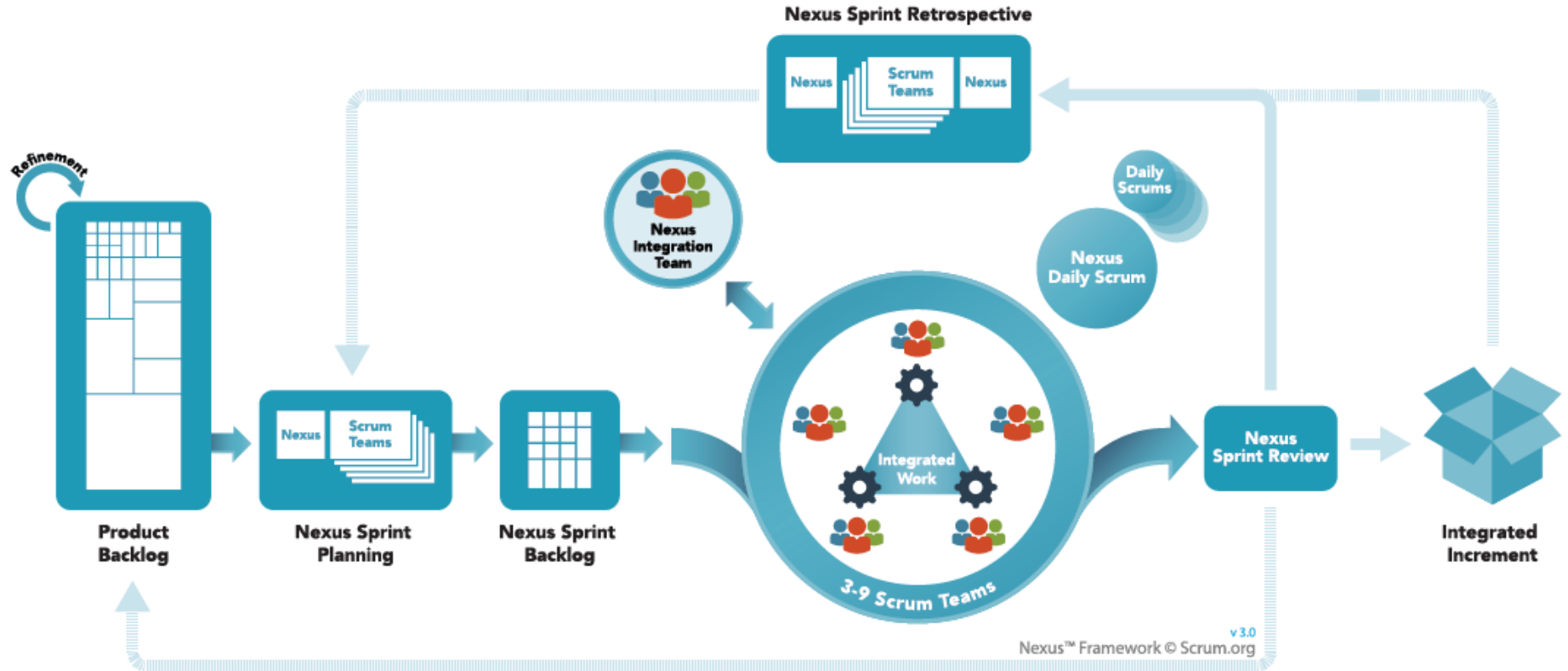


# Итого

Самое простое, что можно сделать

Слишком гибко 😊

# Nexus



# Nexus

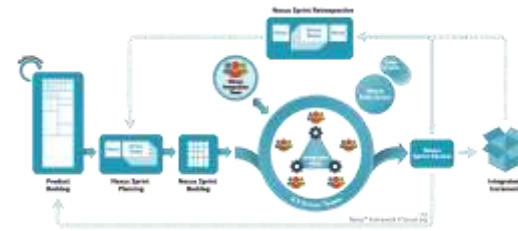
3-9 команд

Один Product Owner

Общий бэклог

Одинаковая длина спринтов

Общий интегрированный инкремент





# Nexus Integration Team



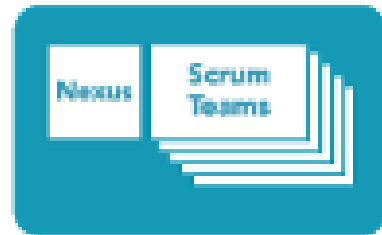
Nexus Integration Team (NIT) – команда, ответственная за

- Успешную интеграцию всех сделанных инкрементов
- Решение вопросов взаимодействия команд

В нее входят представители команд

# Nexus Sprint Planning

- NIT разбивает бэклог на задачи до размера, когда одна команда может закончить задачу за один спринт
- Выявляются и визуализируются зависимости между задачами
- Формируется roadmap: что и какой командой будет сделано в каком спринте
- Команды более подробно анализируют свои задачи



# Nexus Daily Scrum

Аналог Daily Scrum для NIT



- Что было успешно интегрировано до сегодняшнего Daily Scrum?
- Какие новые зависимости обнаружили?
- Какую информацию нужно распространить среди команд сегодня?

# Nexus Retrospective

- NIT определяет проблемы, затронувшие более одной команды
- В командах проводится ретро с учетом общих проблем
- Формируется общее видение как отслеживать сформулированные пункты



# Итого

Тоже классно, когда:

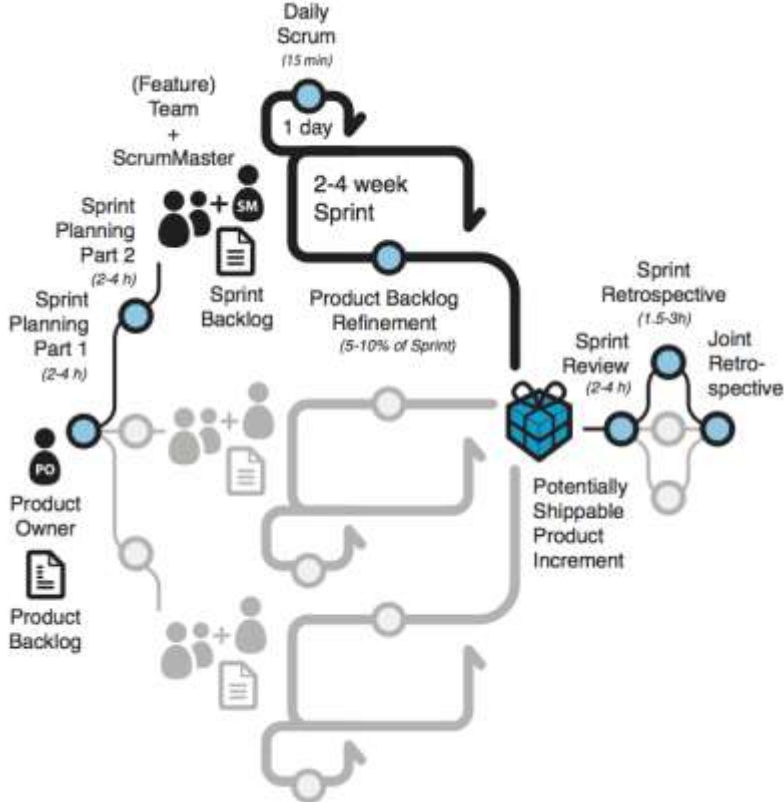
- Команды взаимозаменяемы
- Общий бэклог

Плохо работает, когда:

- Разные бэклоги
- Команды не заменяемые
- Минимизация зависимостей не работает



# Large-Scale Scrum (LeSS)



# Large-Scale Scrum (LeSS)








Отличия от Nexus:

- Планируют инкремент не представители, а команды целиком

Достоинства:


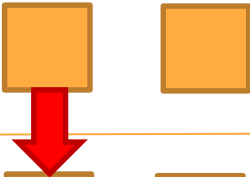




- Есть ресурс с описанием практик
- Есть инфраструктура в виде тренеров и консультантов

# Визуализировать зависимости в спринте

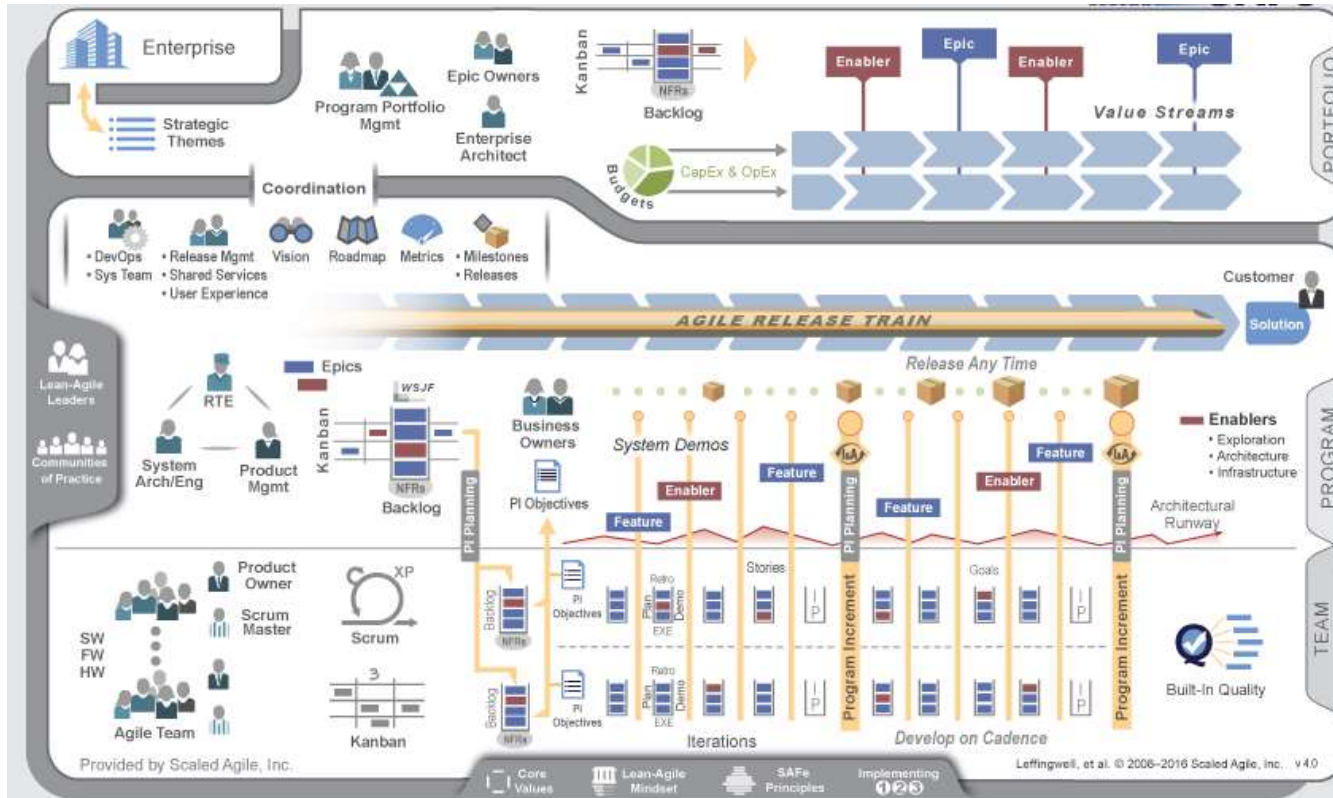
	Blocked	ToDo	In Progress	Done
Team 1				
Team 2				



# Минимизировать зависимости до спринта

	Текущий спринт	Спринт + 1	Спринт + 2
Team 1			
Team 2			

# Scaled Agile Framework (SAFe)



# Scaled Agile Framework (SAFe)

Преимущества:

- Сокращение Time To Market на 30-50%
- Багов меньше на 50%
- 20-50% рост продуктивности
- Более счастливые сотрудники



# Особенности

- 5-12 Agile команд (50-150 человек)
- Команды могут работать по Scrum или Kanban
- Но нужно работать двухнедельными итерациями
- Итерации всех команд выравниваются
- Общий бэклог



# Координация

Роль Release Train Engineer, отвечающий за общий результат

Команда из RTE, Product Manager, System Architect

Еженедельные встречи



# Самое интересное

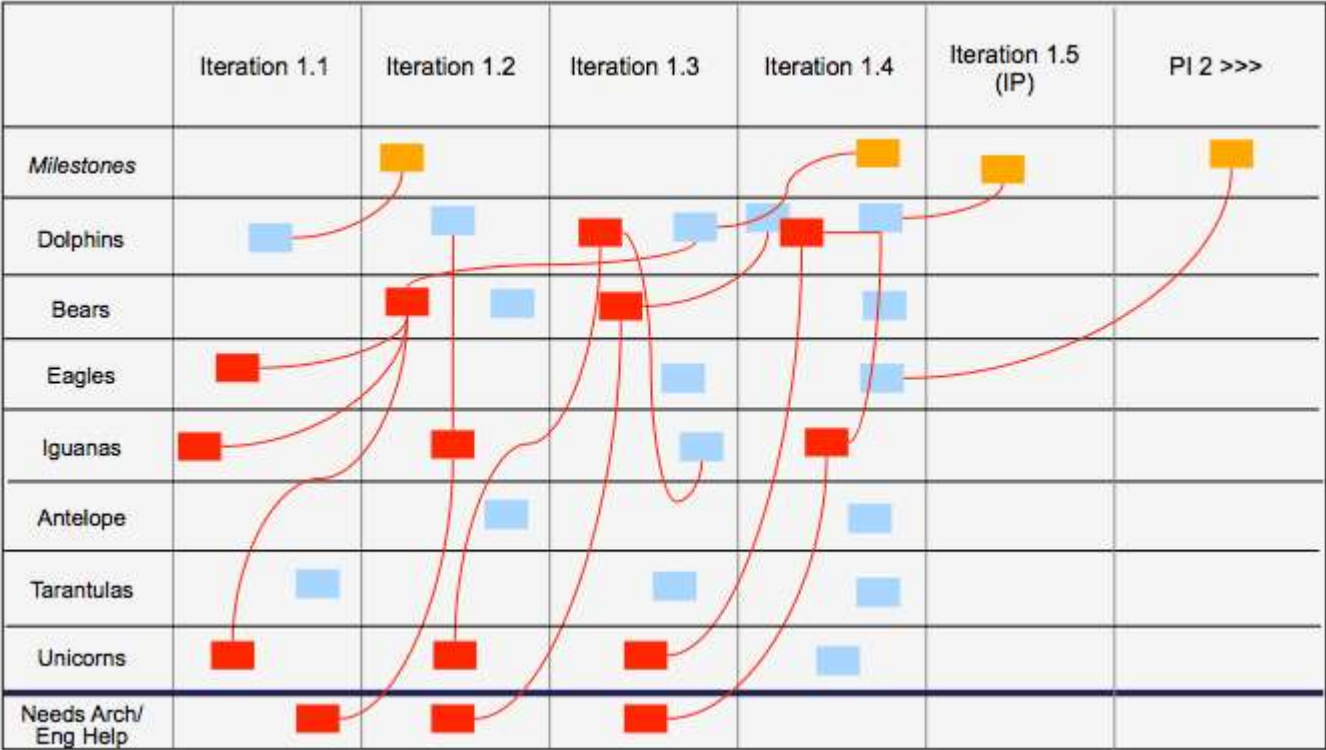
- Релизный цикл (Agile Release Train) – 5 итераций
- В течение цикла возможны промежуточные релизы
- Общий Product Increment (PI)
- PI Planning – планирование PI всеми командами в полном составе!



# PI Panning



# Ключевой момент— связи между задачами





# Кроссфункциональные команды

Dev + QA = Profit

Иначе команды могут держать друг - друга



# Техдолг

Обычно – неприятно, но терпимо

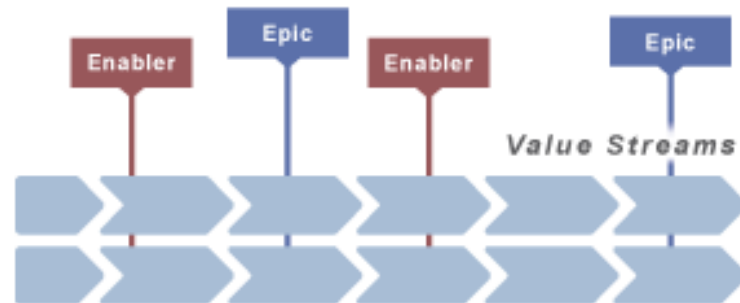
SAFe – препятствие к достижению целей

Enabler – специальная задача для техдолга

Enabler планируется совместно с новыми фичами

Последний спринт ART – прототипы, хакатоны и исследования

Надоело продавать техдолг своему ProductOwner – внедряйте SAFe 😊



# Внедрение

# Воспитать своих аналитиков



Наш ART – месяц (2 итерации по 2 недели)

На момент старта ART - требования на 2 итерации вперед

При анализе фичи прописывается, она нужна:

- в каких продуктах
- для каких кастомеров

По возможности – уже указаны зависимости между задачами

# Унифицировать оценки

- Оценка в одинаковых попугаях
- Заложены одинаковые вещи
- Code Review добавили
- Reverse Integration, Developer testing – убрали
- Одинаково трекаются часы
- Подобная декомпозиция типовых задач
- Добавление поля в программу

0 0	1/2 1/2	1 1	2 2	3 3	5 5
8 8	13 13	20 20	40 40	100 100	? ?

# Унификация мониторинга работы

Все команды работают по скраму

Одинаково настроены доски

Одинаково стартуют/закрываются спринты



# Синхронизовать спринты

Выгоды:

- Если на каком-то проекте понадобится помощь, мы точно знаем, когда к нему может подключиться новая команда
- Есть команда поддержки которая меняется. Нет проблем сделать замену



# Кроссфункциональные команды

Core и API: Dev + QA = Profit

Devops – вне команд

Аналитики – вне команд





# Поддержка в Task tracker

У нас – Jira

- Task и Subtask – в течение спринта
- User Story – в течение ART
- Epic – в течение нескольких ART
  
- Enabler – отдельный Issue Type от Task



# Выводы

- Внедрение SAFe, как и внедрение PMO – это сложно
- Изменение структуры компании
- Быть или не быть QA отделу
- Составление регламентов



# Главное

Внедряйте изменения осознанно! Понимайте, чего хотите добиться

Будет сопротивление, но 5 раз объяснил – сам понял 😊



# Результат

До «перестройки»:

- Любые фичи – от 3 месяцев. Как правило – больше ☹️

После перестройки:

- Мелкие фичи – месяц во всех продуктах
- Большие фичи – 2 месяца во всех продуктах



# Планы

Перейти на ART по 3 месяца через год

- Укладывать в 3 месяца доработку под кастомера
- Синхронизировать работу Dev и Sales



	Январь	Февраль	Март	Апрель	Май	Июнь
Команда 1	Штат А			Штат В		
Команда 2	Программа X	Программа Y				

# Что почитать/посмотреть

Документация по SAFe

<http://www.scaledagileframework.com/>

Ролики

SAFe overview: краткий обзор <https://www.youtube.com/watch?v=NOcYhwarLkM>

SAFe Portfolio & Value stream <https://www.youtube.com/watch?v=1Kv9-qiYFQM>

SAFe Agile Release Train execution <https://www.youtube.com/watch?v=utDldiQdoqU>

SAFe Практика внедрения на реальных проектах <https://www.youtube.com/watch?v=G1ZPndN8dto&t>

Цветчих Денис

Руководитель проектов в компании OneSystems

[den.tsvettsih@yandex.ru](mailto:den.tsvettsih@yandex.ru)

+7 923 272 32 48

