

Microsoft Solutions Framework

Модель процессов MSF

Источники информации

- *Майкл С. В. Тернер. Основы Microsoft Solution Framework, СПб.:Питер, 2008 – 336с.*
- [Azure DevOps](#) – предоставляет командам разработчиков средства для планирования, совместной работы по разработке кода, а также для создания и развертывания приложений (на смену Team Foundation Server)
- [MSF для гибкой разработки программного обеспечения версии 5.0](#)
- [MSF for Agile Software Development 6.0](#)
- [Agile-манифест разработки программного обеспечения](#)
- [Agile Guide](#)

Microsoft Solutions Framework (MSF)

Методология разработки программного обеспечения от Microsoft.

- MSF описывает управление людьми и рабочими процессами в процессе разработки решения.

MSF состоит из двух моделей и трех дисциплин.

- модели:

- ☐ модель проектной группы
- ☐ модель процессов

- дисциплины:

- ☐ дисциплина управление проектами
- ☐ дисциплина управление рисками
- ☐ дисциплина управление подготовкой

Модель проектной группы MSF

- **Описывает** подход Майкрософт к организации работающего над проектом персонала и его деятельности в целях максимизации успешности проекта.
- **Определяет** ролевые кластеры, их области компетенции и зоны ответственности, а также рекомендации членам проектной группы, позволяющие им успешно осуществить свою миссию по воплощению проекта в жизнь.
- **Включает** в себя ряд основных принципов, которые имеют отношение к успешной работе команды:
 - Распределение ответственности при фиксации отчетности
 - Наделение членов команды полномочиями

Модель проектной группы



Функциональные группы

- Функциональные группы – это подкоманды, существующие внутри ролевых кластеров.
 - формируются, когда стоящие перед ролевым кластером задачи столь масштабны, что требуют выделения специальных ресурсов

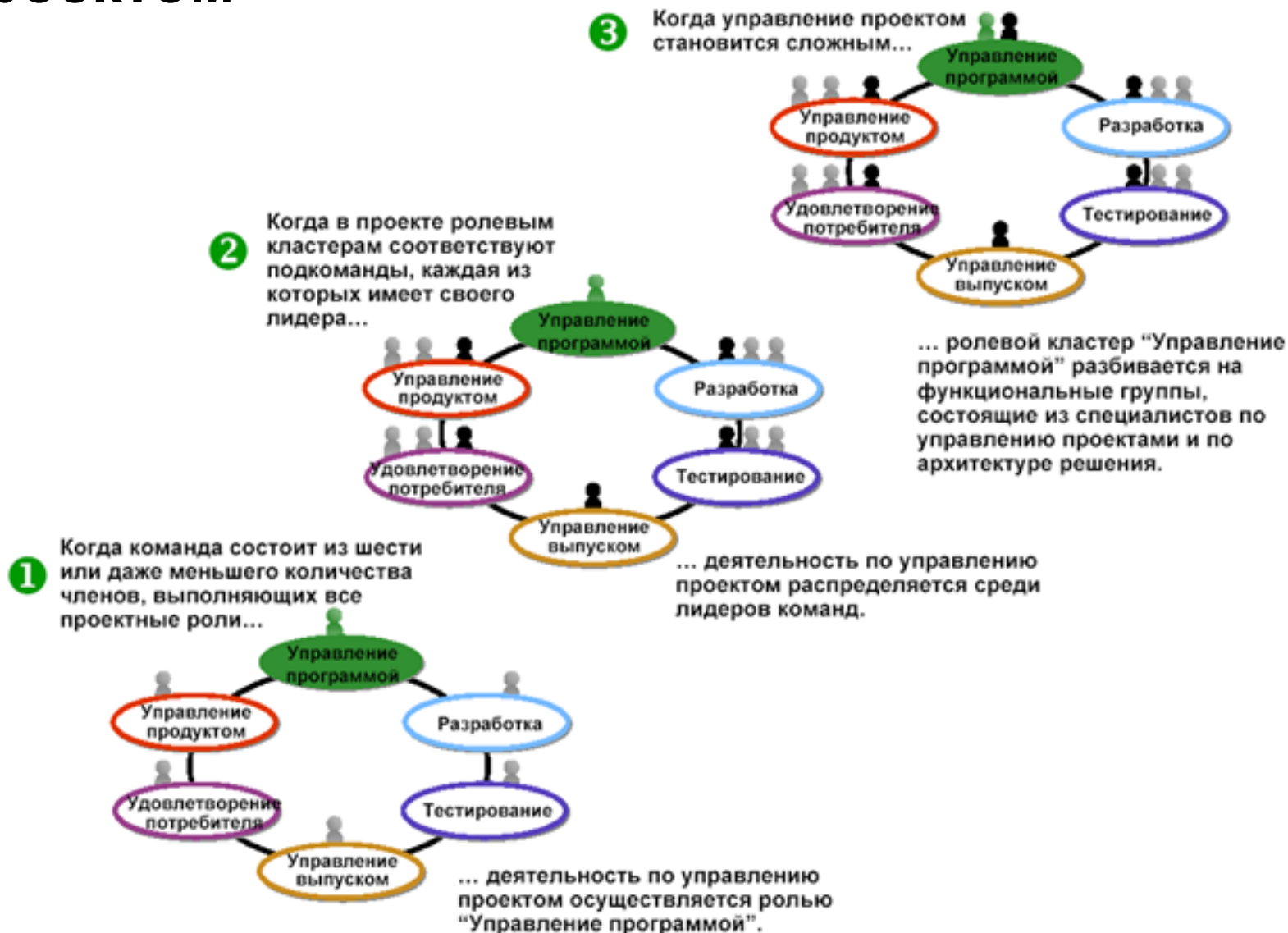


Группы направлений

- Группы направлений – это многопрофильные подкоманды, организуемые для создания определенной составляющей решения.
 - komponуются из ролей модели проектной группы.



Масштабирование функций управления проектом



Модель процессов MSF

Модель процессов (process model) представляет общую методологию разработки и внедрения IT-решений.

Особенности модели:

- может быть применена при разработке широкого круга IT-проектов
- модель сочетает в себе свойства двух стандартных производственных моделей: каскадной (waterfall) и спиральной (spiral)
- процесс ориентирован на “вехи” (milestones) – ключевые точки проекта, характеризующие достижение в его рамках какого-либо существенного (промежуточного либо конечного) результата
- модель процессов MSF учитывает постоянные изменения проектных требований

Базовые принципы MSF

- Единое видение проекта
 - для этой цели специальная фаза (“Выработка концепции”), которая заканчивается вехой
- Проявляйте гибкость – будьте готовы к переменам
 - принцип непрерывной изменяемости условий проекта при неизменной эффективности управленческой деятельности
- Концентрируйтесь на бизнес-приоритетах
 - модель процессов включает в свой жизненный цикл не только разработку продукта, но и его внедрение
- Поощряйте свободное общение
 - модель процессов предлагает проведение анализа хода работы над проектом в определенных точках

Ключевые термины модели процессов MSF

- “заказчик” (customer) и “потребитель” (пользователь, user) продукта
- заинтересованные стороны (stakeholders)
- “решение” (solution)
- базовая версия (baseline)
- рамки (scope)
 - рамки решения
 - рамки проекта

Что есть решение?

- **“Решение”** (solution) - *скоординированная поставка* набора элементов (таких как программно-технические средства, документация, обучение и сопровождение), необходимых для удовлетворения некоторой бизнес-потребности *конкретного заказчика*
- Решение может включать в себя один или несколько программных продуктов, тем не менее, нужно четко разграничивать продукты и решения



Продукты и решения

■ Продукты

- ☐ Разрабатываются для нужд массового рынка.
- ☐ Поставляются в качестве дистрибутивных пакетов или загружаемых файлов.

■ Решения MSF

- ☐ Разрабатываются или привязываются к нуждам определенного заказчика.
- ☐ Поставляются путем внедрения проекта.

Элементы успешного решения



Рамки проекта и рамки решения

- *Рамки решения (solution scope)* определяют функциональность решения и его возможности (включая те, что не относятся к программному обеспечению).
- *Рамки проекта (project scope)* определяют объем работ, который должен быть выполнен проектной группой для поставки заказчику каждого из элементов, определенного рамками решения.



Ключевые концепции модели процессов MSF

- Создание базовых версий
 - версия (baseline) – это известное и зафиксированное состояние чего-либо, используемое для последующего сравнения
- Управление компромиссами

Треугольник компромиссов



Ограничения

- Стоимость
- Время
- Функции

После достижения равновесия в этом треугольнике изменение на любой из его сторон для поддержания баланса требует модификаций на другой (двух других) сторонах и/или на изначально измененной стороне.

Пример. Если фиксирован бюджет – варьируйте функции

Матрица компромиссов проекта



	Фиксируется	Согласовывается	Принимается
Ресурсы			
Время			
Возможности			



Характеристики модели процессов MSF

- Подход, основанный на фазах и вехах.
- Итеративный подход.
- Интегрированный подход к созданию и внедрению решений.

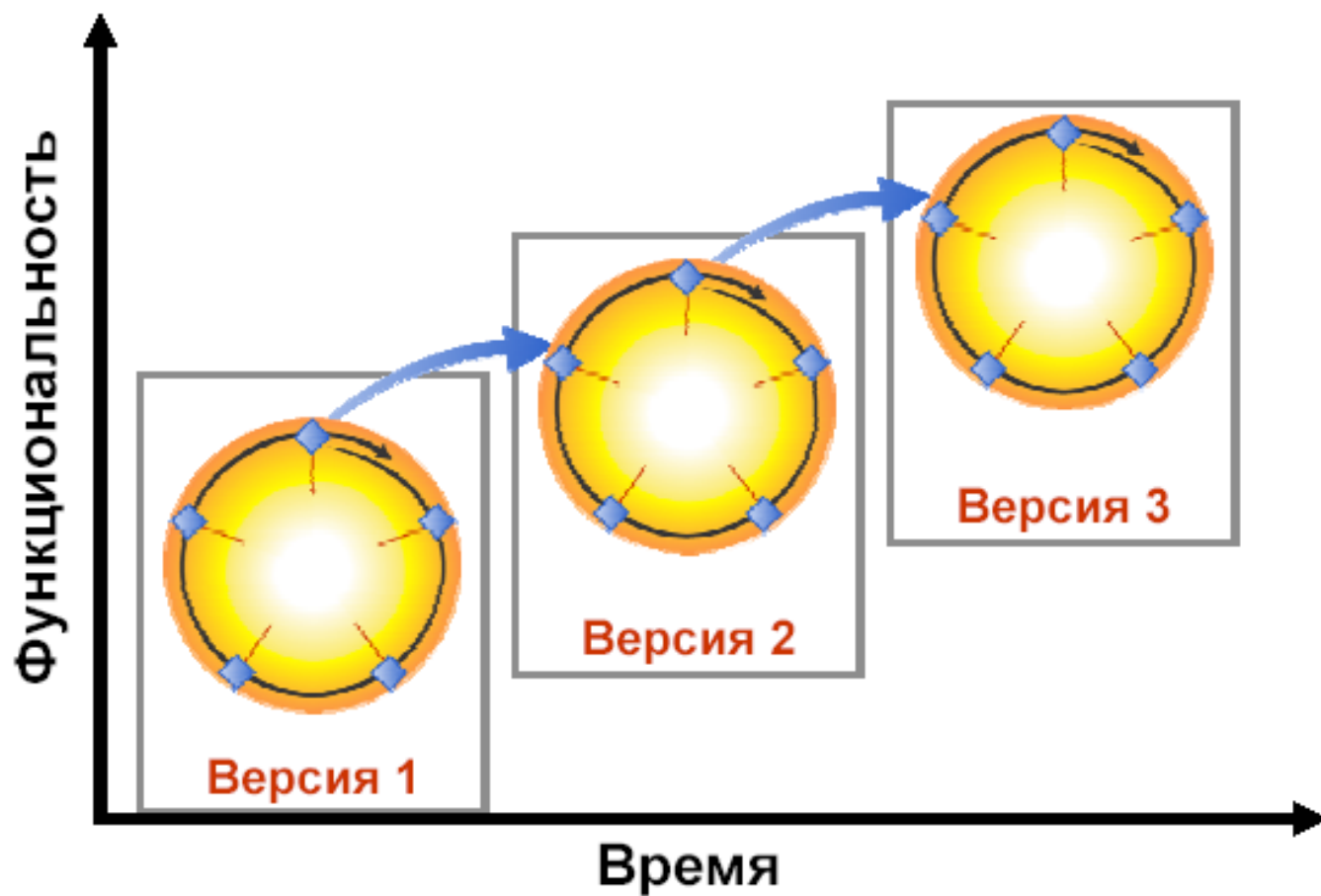
Подход, основанный на вехах


- Вехи - опорные точки для планирования и мониторинга хода проекта:
 - главные (major) - точки перехода от одной фазы к другой
 - промежуточные (interim) - показывают достижение в ходе проекта определенного прогресса и расчленяют большие сегменты работы на меньшие, обозримые участки
- Вехи - точки синхронизации
- Вехи - ориентиры производственной ответственности

Ведущие роли различных фаз

Веха	Ведущие ролевые кластеры
Концепция утверждена	Управление продуктом
Планы проекта утверждены	Управление программой
Разработка завершена	Разработка, Удовлетворение потребителя
Готовность решения утверждена	Тестирование, Управление выпуском
Внедрение завершено	Управление выпуском

Итеративный подход





Характеристики итеративного подхода

- Выпуск версий
- Создание “живой” документации
- Ранние базовые версии, отложенные итоговые версии
- Ежедневные билды
- Управление конфигурациями проекта



Рекомендации для выпуска версий решения

- Создавая планы, предусматривайте версионирование.
- Прежде всего, поставляйте базовую функциональность.
- Выбирайте приоритеты, учитывая риски.
- Осуществляйте частые итерации разработки.
- Формализуйте процедуры контроля изменений в проекте
- Не создавайте новых версий, если они не увеличивают ценность решения.

Интегрированный подход к созданию и внедрению решений

■ Фазы и вехи модели процессов MSF



Фаза выработки концепции (envisioning)


- Концепция должна включать содержательное описание целей продукта и предполагаемый результат.
- Концепция должна описывать решаемую проблему и перспективу ее решения.
 - ☐ включает описание задачи,
 - ☐ ее возможности,
 - ☐ решения и преимущества
- Концепция должна пробуждать интерес клиентов и членов команды

Фаза выработки концепции (envisioning)

- Создание и сплочение проектной группы на основе выработки единого видения.
- Основными задачами фазы являются:
 - Создание ядра проектной группы
 - Подготовка документа общего описания и рамок проекта (vision/scope document).
- Проектная группа готовит документ оценки рисков и представляет главные риски проекта вместе с общим описанием и рамками проекта
- Производится выявление и анализ бизнес-требований.

Вехи фазы выработки концепции и результаты

- **главная веха:**
 - **Веха “Концепция утверждена”**
- **рекомендуемые промежуточные вехи:**
 - Ядро проектной группы сформировано
 - Черновой вариант концепции проекта составлен
- **Результаты:**
 - Общее описание и рамки проекта (vision/scope document).
 - Документ оценки рисков (risk assessment document).



Фаза планирования (planning)

Основная работа по составлению планов проекта:

- подготовка проектной группой функциональной спецификации,
- разработка дизайнов,
- подготовка рабочих планов,
- оценка проектных затрат и сроков разработки различных составляющих проекта

Вехи фазы планирования и результаты

- главная веха:
 - ***Веха “Планы проекта утверждены”***
- рекомендуемые промежуточные вехи:
 - Верификация технологий
 - Базовая версия функциональной спецификации создана
 - Базовая версия сводного плана проекта создана
 - Базовая версия сводного календарного графика проекта создана
 - Среды разработки и тестирования развернуты
- Результаты:
 - Функциональная спецификация.
 - План управления рисками.
 - Сводный план и сводный календарный график проекта



Фаза разработки (developing)

Задачи:

- создание компонент решения (включая как документацию, так и программный код).
- разработка инфраструктуры.

Вехи фазы разработки и результаты

- главная веха:
 - **Веха “Разработка завершена”**
- рекомендуемые промежуточные вехи:
 - Концепция подтверждена
 - Билд n завершен, билд n+1 завершен...
- Результаты:
 - Исходный и исполнимый код приложений.
 - Скрипты установки и конфигурирования.
 - Окончательная функциональная спецификация.
 - Материалы поддержки решения.
 - Спецификации и сценарии тестов.



Фаза стабилизации (stabilizing)

Производятся работы:

- тестирование разработанного решения
- устранение ошибок
- подготовка решения к выпуску

Вехи фазы стабилизации

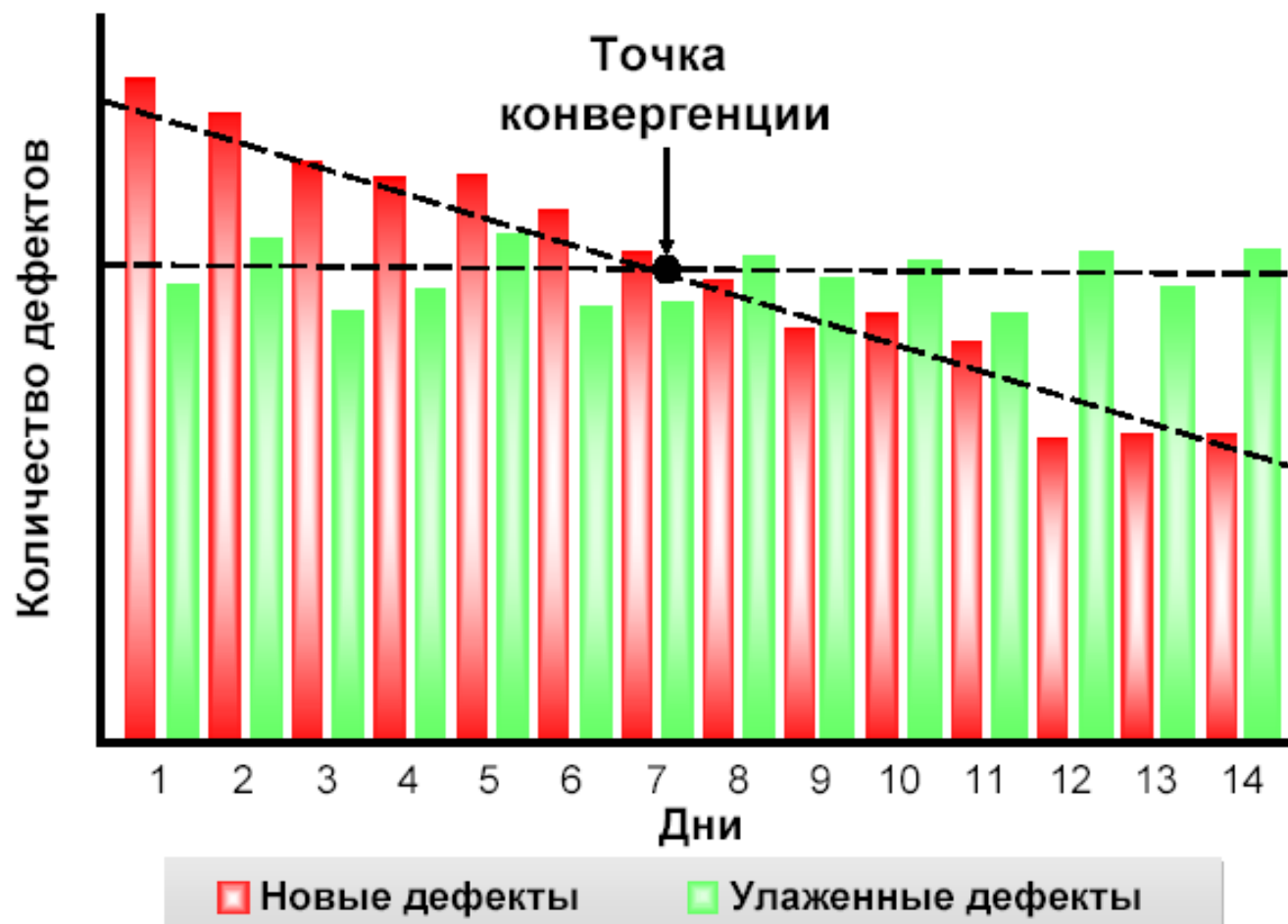
- главная веха:
 - ***Веха “Готовность решения утверждена”***
- рекомендуемые промежуточные вехи:
 - Точка конвергенции
 - Точка достижения нуля
 - Версии-кандидаты
 - Контрольное тестирование завершено
 - Тестирование приемлемости для потребителей завершено
 - Пилотное внедрение завершено



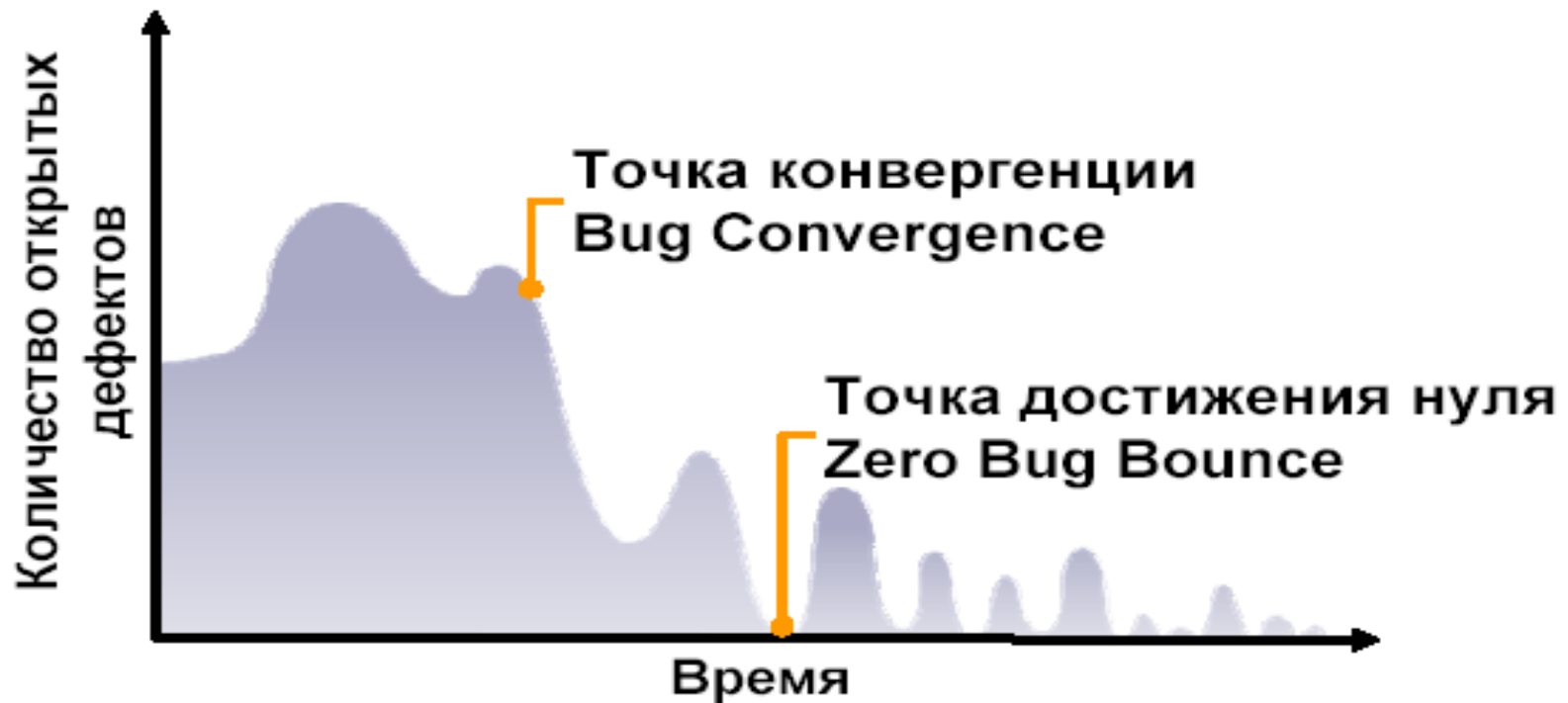
Результаты фазы стабилизации

- Окончательный продукт (golden release).
- Документация выпуска (release notes).
- Материалы поддержки решения.
- Результаты и инструментарий тестирования.
- Исходный и исполнимый код приложений.
- Проектная документация.
- Анализ пройденной фазы (milestone review).

Точка конвергенции



Точка достижения нуля





Фаза внедрения

Работы:

- внедрение технологии и компонент решения,
- стабилизация внедренного решения,
- передача работы персоналу поддержки и сопровождения
- получение со стороны заказчика окончательного одобрения результатов проекта.
- могут продолжаться меры по стабилизации решения

По завершению внедрения:

анализ выполненной работы и удовлетворенности заказчика.

Вехи фазы внедрения

- главная веха:
 - ***Веха “Внедрение завершено”***
- рекомендуемые промежуточные вехи:
 - Ключевые компоненты развернуты
 - Внедрение на местах завершено
 - Внедренное решение стабилизировано



Результаты фазы внедрения

- Информационные системы эксплуатации и поддержки.
- Процедуры и процессы.
- Базы знаний, отчеты, журналы протоколов (logbooks).
- Версии проектных документов, массивы данных (load sets) и программный код, разработанные во время проекта.
- Отчет о завершении проекта (project close-out report).
- Окончательные версии всех проектных документов.
- Показатели удовлетворенности заказчика и потребителей.
- Описание последующих шагов.

Рекомендуемые методики модели процессов MSF

- Стимулируйте изобретательность расширяя функциональность и ограничивая ресурсы
- Фиксируйте календарный график
- Календарное планирование на неопределенное будущее
- Используйте параллельно работающие компактные команды
- Разбивайте большие проекты на осуществимые части
- Извлекайте уроки из пройденных вех
- Используйте прототипирование
- Используйте частые билды и быстрые тесты
- Частые итерации разработки и внедрения
- Избегайте расползания рамок проекта
- Оценка снизу вверх
- Интегрирование представленных проектной группой оценок

MSF 4.0

- Версия MSF 4.0 была представлена в 2005 году.
- Произошло разделение методологии на два направления:
 - MSF for Agile Software Development - **ориентируется на небольшие** команды (5-6 человек), предполагает, что информация о разрабатываемом продукте не просто выясняется в процессе разработки, а может и будет изменяться по ходу.
 - MSF for CMMI Process Improvement - строгий, документированный процесс, рассчитанный на **большие** команды и **длительный** процесс разработки, что предполагает больше верификации, больше планирования, процедуры утверждения, отслеживание потраченных ресурсов и т.д.

Основные положения MSF for Agile Software Development

- Первая рабочая версия системы должна быть создана как можно раньше, а сам продукт фактически проявляется из прототипов путем повторения итераций в цикле разработки.
- Методология MSF содержит ряд элементов, в частности:
 - рекомендованные процессы создания IT-проектов;
 - структуру итераций;
 - роли членов команды;
 - шаблоны документов (Excel, Word);
 - шаблоны Microsoft Project;
 - отчеты;
 - портал проекта (шаблон сайта SharePoint).
- MSF for Agile Software Development ориентирован на использование итеративной и эволюционной модели процесса разработки



Модель проектной группы MSF for Agile Software Development

Основные принципы построения команды

- Концентрация на нуждах заказчика
- Нацеленность на конечный результат
- Установка на отсутствие дефектов
- Проектная группа - команда равных
- Стремление к самосовершенствованию

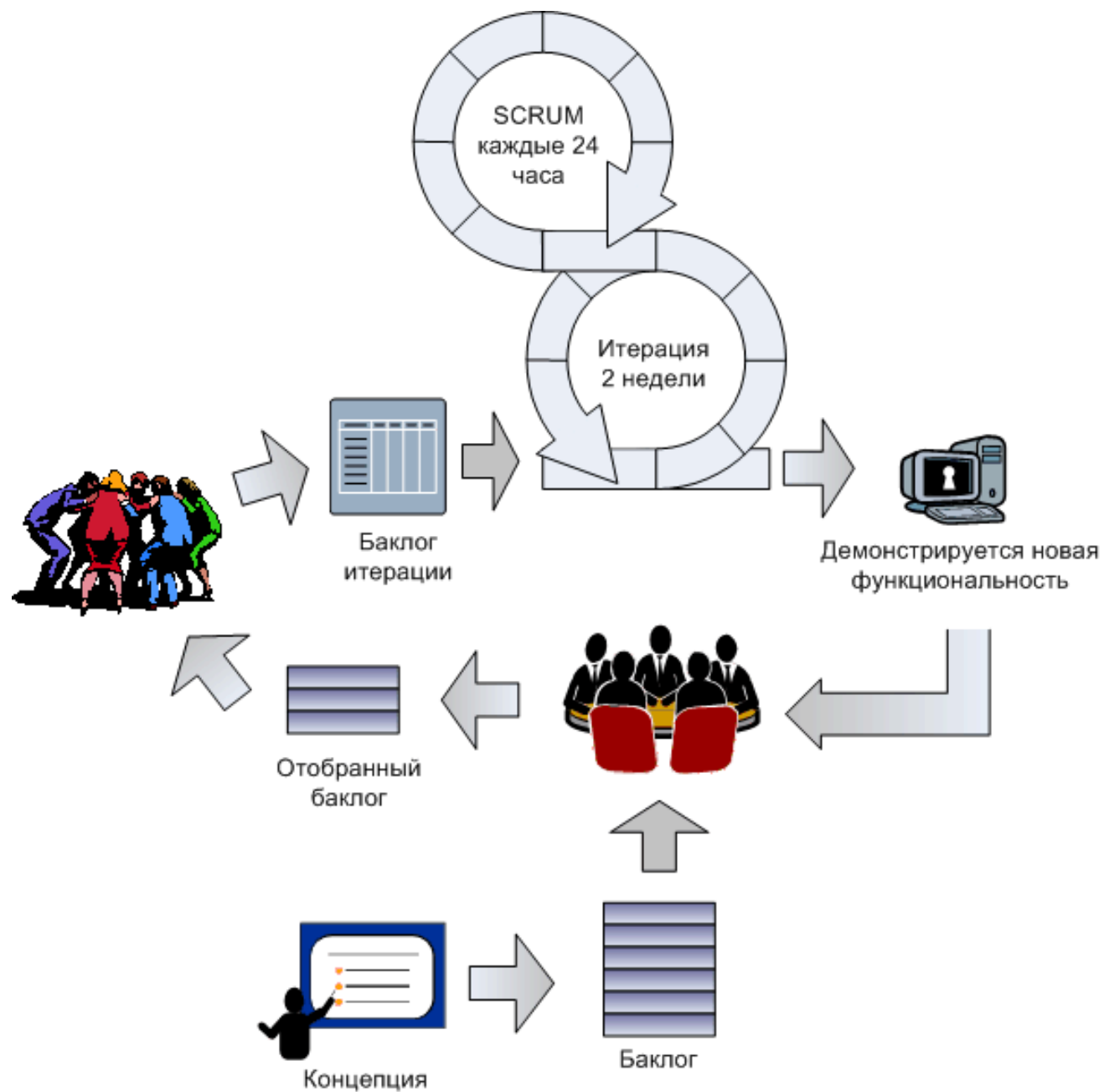
MSF 5.0 для гибкой разработки ПО

- **Scrum.** Scrum — платформа для управления разработкой сложных продуктов и систем, характеризующаяся гибкими принципами и характеристиками.
- **Рекомендации по проектированию.** Эти рекомендации помогают увеличить скорость, с которой команда предоставляет желаемые результаты клиентам.
- **Артефакты.** Каждый артефакт служит для реализации определенной функции и предоставляет возможности для уточнения процессов с течением временем.
- **Роли.** В процессе Scrum определены три роли.
 - Scrum Master – мастер, координатор,
 - Product Owner – владелец продукта
 - Team – команда
- **Собрания.** Проводится ряд собраний. Каждое собрание имеет конкретную цель, проводится с определенной периодичностью и ограничено по времени.

Проблемы, которые решаются запуском Scrum-команд

- Проекты делаются долго, некоторые проекты даже не начинаются.
- Медленная работа подразделения и низкая вовлеченность сотрудников в процесс.
- Долгое и неэффективное взаимодействие между подразделениями.
- Процессы идут активно, только если участвуют руководители.

Scrum



Спринт (Sprint)

- Sprint – итерация (цикл выпуска продукта):
 - имеет фиксированную длительность, обычно от двух до восьми недель, в течение которых выполняются все действия по разработке
 - результат – готовый продукт (build), который можно потенциально передавать заказчику
- Каждый спринт – маленький "водопад":
 - в течение спринта делаются все работы по сбору требований, дизайну, кодированию и тестированию продукта.
- Рамки спринта должны быть фиксированными.
- Каждый спринт начинается с собрания по его планированию и заканчивается собранием, где подводятся итоги спринта.

Спринт (Sprint)

- Структура спринта

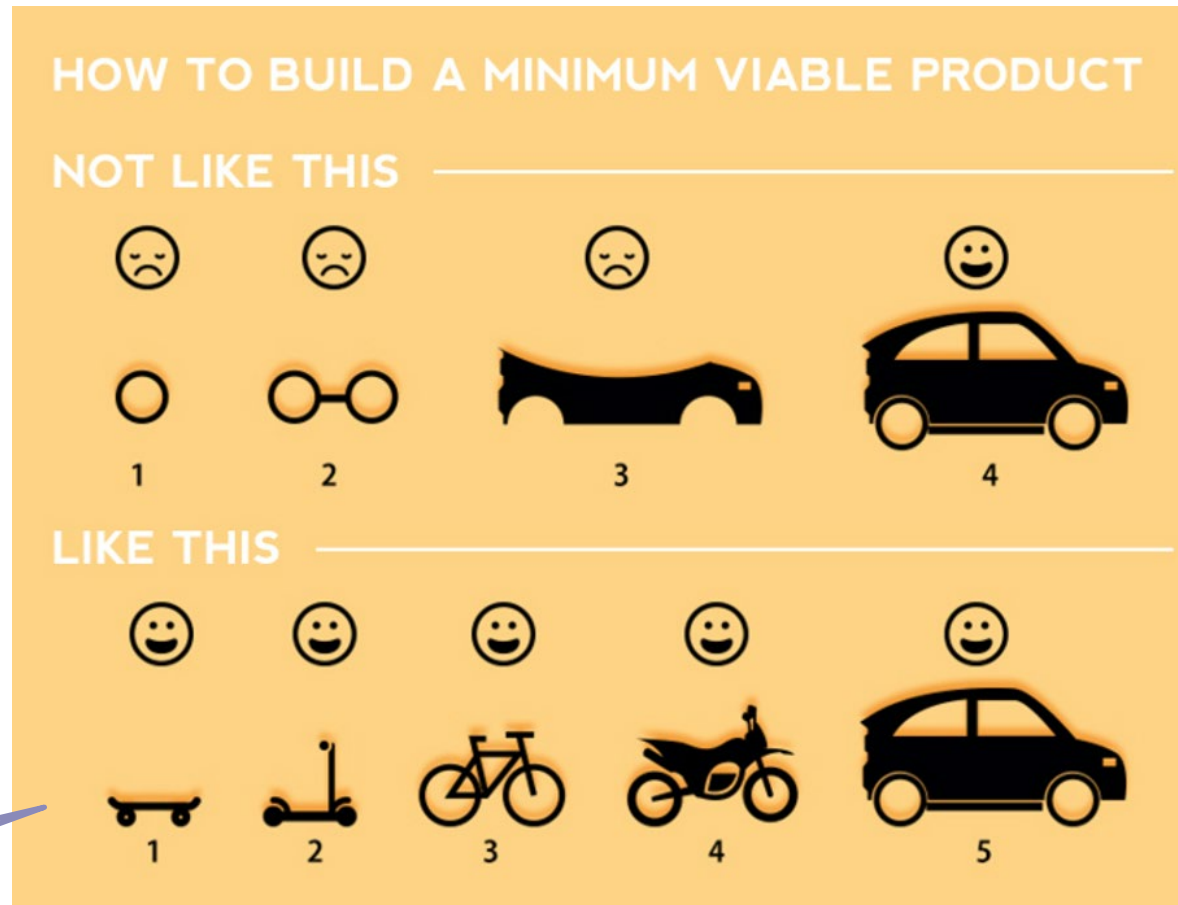


Agile-подход к разработке продукта

После каждого спринта продукт должен быть:

- работоспособным
- полезным для пользователя
- более совершенным, чем до спринта

минимально
работоспособный
продукт



С каждым спринтом продукт
становится полезнее



Рекомендации по проектированию

■ Непрерывное построение и развертывание

- Если команда чаще возвращает код в систему управления версиями и выполняет построения, обычно это приводит к увеличению скорости работы команды.

■ Раннее и частое тестирование

- Команда должна выполнять раннее тестирование, а также часто тестировать код по мере его построения.

■ Моделирование приложения

- Модели можно использовать для анализа и рефакторинга существующего кода, эффективного изучения требований клиентов, определения и описания структуры программного обеспечения и получения сведений, необходимых при разработке приемочных тестов и тестов компонентов.

Артефакты

- Для упрощения процессов, реализуемых координаторами Scrum, и повышения эффективности работы команды.
 - **рабочие элементы** – для отслеживания данных, анализа хода выполнения и принятия решений
 - **отчеты**, основанные на базе данных – для отслеживания рабочих элементов или данных служб аналитики SQL Server
 - **книги** – для ведения учета невыполненной работы по продукту, планирования итераций или спринтов и назначения приоритетов ошибок
 - **панели мониторинга** – для отображения важной информации и обеспечения прозрачности и актуальности показателей


Роли

- Scrum Master - координатор команды
 - отвечает за создание эффективной команды и организацию ее работы в соответствии со спецификой Scrum-процессов
- Product Owner – владелец продукта (“голос клиента”)
 - отвечает за разработку продукта, определяет какие функции реализуются в продукте
 - ставит задачи команде, но он не вправе ставить задачи конкретному члену проектной команды в течение спринта
- Team – команда
 - берет на себя обязательства по выполнению объема работ на спринт перед Product Owner.
 - работа команды оценивается как работа единой группы. В Scrum вклад отдельных членов проектной команды не оценивается, так как это разваливает самоорганизацию команды

Собрания

- При использовании командой платформы **Scrum** проводится серия собраний, каждое с определенной целью и частотой

Собрание	Предназначение	Длительность	Частота
Собрание по планированию спринта	Определение необходимых работ для ближайшего спринта.	Два часа на неделю спринта, до четырех часов	Один раз для каждого спринта
Ежедневное Scrum-собрание	Позволяет участникам команды принимать, разделять и обмениваться рисками.	Пятнадцать минут	ежедневно
Краткое собрание по проверке результатов	Демонстрация клиенту и другим заинтересованным лицам выполненной командой работы в спринте и ознакомление с отзывами о ней.	Два часа на неделю спринта, до четырех часов	Один раз для каждого спринта
Ретроспективное собрание	Определение и внедрение идей для усовершенствования процесса.	Три часа	Один раз для каждого спринта



Стивен Деннинг Эпоха Agile. Как умные компании меняются и достигают результатов. — «Манн, Иванов и Фербер (МИФ)», 2018

- Аарон Бьйорк, менеджер группы проектов в подразделении разработки Microsoft:
 - история об Agile-трансформации Microsoft
- Ключевые аспекты, необходимые для внедрения Agile в глобальном масштабе
 - На следующих трех слайдах

Ключевые аспекты, необходимые для внедрения Agile в глобальном масштабе

- Найти баланс между согласованностью и автономностью
 - Обеспечить согласованность наверху и автономность внизу
- Освоить роль Agile-менеджера
 - Команда должна понимать, почему следует практикам и несет ответственность за созданную ценность
- Управлять отношениями на уровне команды
 - Внутренние отношения регулируются самими командами настолько, насколько это возможно, но каждая знает, чем занимаются остальные
- Обеспечить постоянную интеграцию
 - нельзя работать автономно в течение трех недель в надежде объединить результаты позже — необходимо ежедневное сотрудничество

Ключевые аспекты, необходимые для внедрения Agile в глобальном масштабе

- Следить за техническим долгом
- Внедрить DEVOPS и постоянную поставку
 - При методе DevOps разработка и эксплуатация объединяются. Команды занимаются планированием, выполнением, поставкой и рабочими процессами для каждой новой функции
- Постоянно отслеживать прогресс
 - Команды постоянно отслеживают, как функции используются в данный момент. Результаты отражаются в бэклоггах, которые называются сценариями.
- Прислушиваться к желаниям пользователей, но удовлетворять их потребности
 - Программные менеджеры изучают своих пользователей всевозможными способами

Ключевые аспекты, необходимые для внедрения Agile в глобальном масштабе

- Использовать самоформирующиеся команды, чтобы вдохновить людей на коллективную ответственность
 - Менеджеры позволяют сотрудникам периодически выбирать команду для работы
- Признать, что команда — это продукт
 - На практике активом все чаще и чаще является команда, способная поставлять программные продукты
- Создавать качество с самого начала
 - Правила предписывают: поставлять готовый продукт после каждого спринта
- Использовать коучинг с осторожностью
 - Стандартной образовательной программы нет и то, что работает в других компаниях, может не подойти вам

Agile-манифест

Agile-манифест — это документ, который определяет ценности agile и содержит четыре фундаментальные идеи:

- **Люди и взаимодействия** важнее *процессов и инструментов*
- **Работающие продукты** важнее *исчерпывающей документации*
- **Сотрудничество с заказчиком** важнее *проработки деталей контракта*
- **Готовность к изменениям** важнее *следования первоначальному плану*

Agile не отрицает важности того, что справа, но
больше ценит то, что слева

Agile - от А до Я

с добавками идей из Lean UX



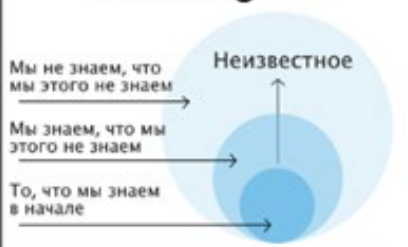
Мы постоянно открываем для себя более совершенные методы разработки программного обеспечения, непосредственно занимаясь разработкой и помогая в этом другим.

Благодаря этой работе мы смогли осознать следующие ценности:
Люди и взаимодействие вместо процессов и инструментов
Работающий продукт вместо исчерпывающей документации
Сотрудничество с заказчиком вместо согласования контрактов
Ответ на изменения вместо следования плану

Манифест гибкой разработки, 2001

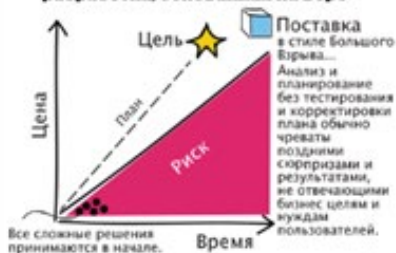
Версия 2.4-ru-1

Зачем Agile



Waterfall

- разработка, основанная на вере



Agile

- разработка, основанная на результатах



Инфографика
Mia Kolmodin
DANDY PEOPLE
blog.dandypeople.com

Перевод
Алексей Кривизный
SCRUM
Украина
www.scrum.ua/materials

Водопадный подход



Подходы к работе



Командная работа для поиска ценности



Modern Agile



Инкрементальная & Итеративная разработка

	1	2	3
GUI			
Клиент			
Сервер			
БД			

Беклог продукта



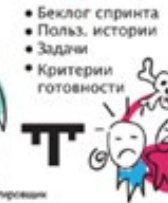
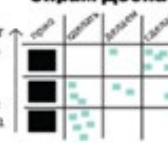
Владелец Продукта (PO) владеет беклогом продукта и видением.

Скрам Команда

Крутая кросс-функциональная команда общается живую, имеет мандат на принятие решений по ценности для бизнеса и пользователя, а также на технические решения. Имеет все навыки для разработки и выпуска готового продукта.



Скрам Доска



Спринты

- Agile-пулс / Постоянный ритм
- Проработка беклога для поиска ценности
- Планирование спринта
- Цель спринта
- Ежедневный митинг 15 мин
- Ретроспектива
- Командные договорённости



Быть Agile



Synefin /культуры/



Agile Onion от AWA, Simon Powers
Synefin от Dave Snowden
Modern Agile от Joshua Kerievsky

