Администрирование информационных систем

Смирнов Михаил
СПбГУ
2009

Процессы разработки

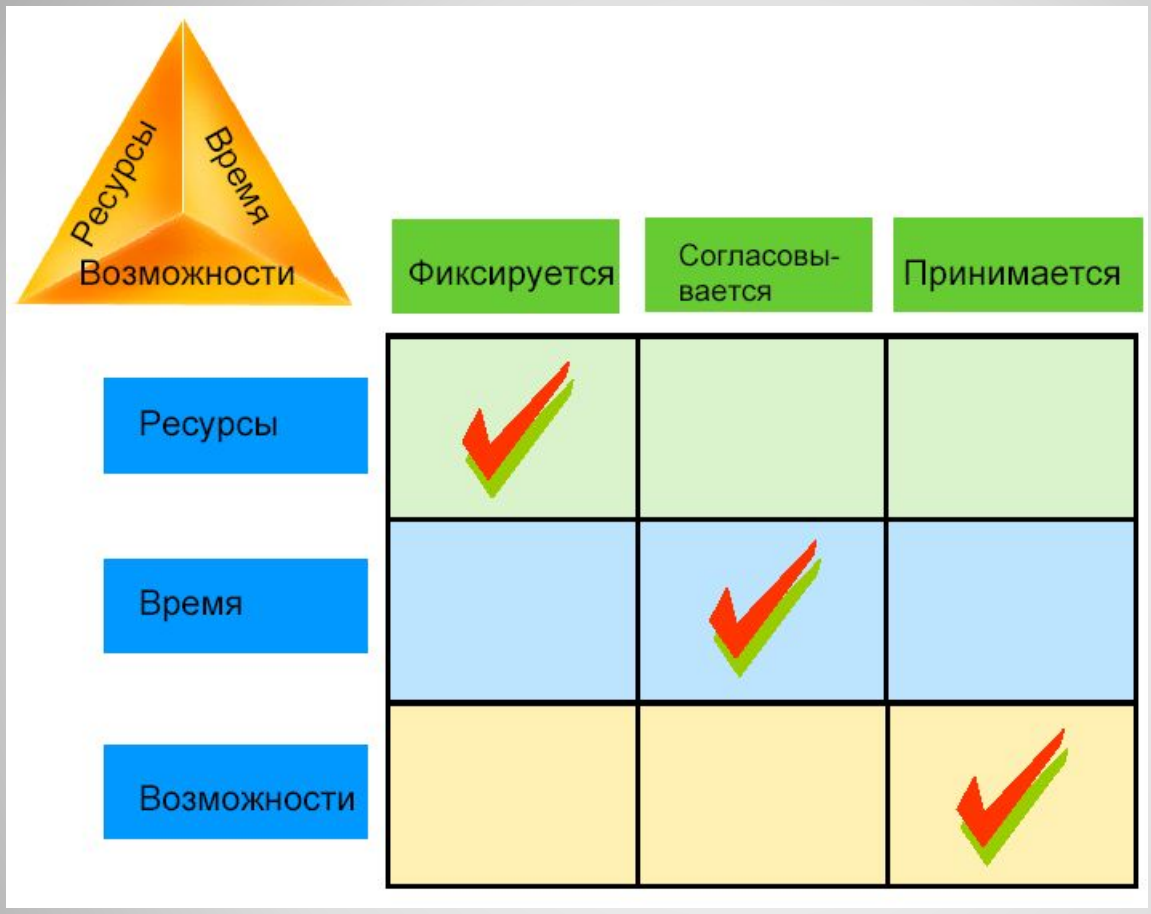
- MSF
- RUP
- Agile
  - XP
  - Lean
  - Scrum

Введение в MSF 3.1

- MSF (Microsoft Solutions Framework) – методология разработки IT-решений, применяемая в Microsoft с 1991 г.
- Мы рассмотрим версию 3.1 (2001 г.)
- Современная версия 4.0 (2005 г.), интегрирована в Visual Studio

Треугольник компромиссов

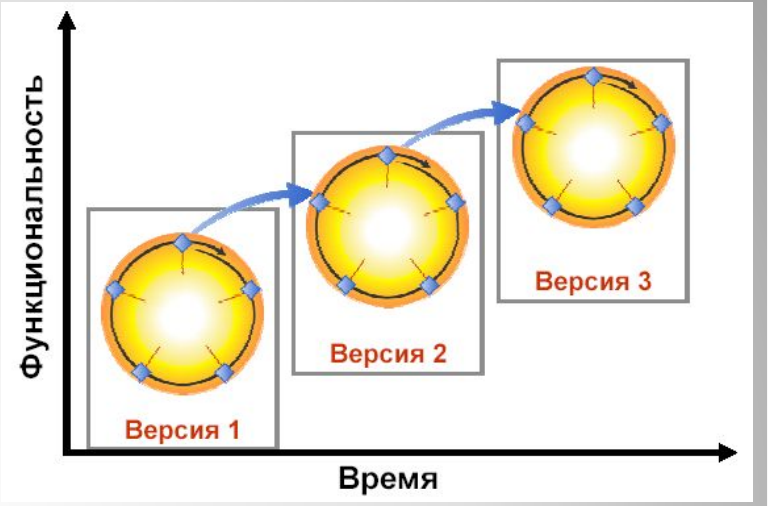
- Характеристики процесса разработки тесно связаны
- Изменение одной из них влияет на другие
- Зафиксировать все 3 характеристики невозможно
- Иногда добавляют дополнительные измерения – качество, процесс разработки и т.п.



Матрица компромиссов

Модель процесса: основы

- Baseline early, freeze late
- Выполняйте ежедневные сборки
- Планируйте версионирование
- Начинайте с базовой функциональности



- Выбирайте приоритеты, учитывая риски Частые итерации и взаимодействие с заказчиком
- Осуществляйте строгий контроль изменений

Модель процесса: фазы и вехи

- Виды деятельности распределены по разным фазам

- Ролевые кластеры команды равных (team of peers)

Модель команды

Масштабирование команды

- В команде выделяются группы по направлениям

- Применяется в offshore-проектах

- Риск – это событие или условие, которое может оказать позитивное либо негативное влияние на итоги проекта
- Риск - неотъемлемая часть всякого проекта или процесса
- Наиболее эффективно превентивное управление рисками

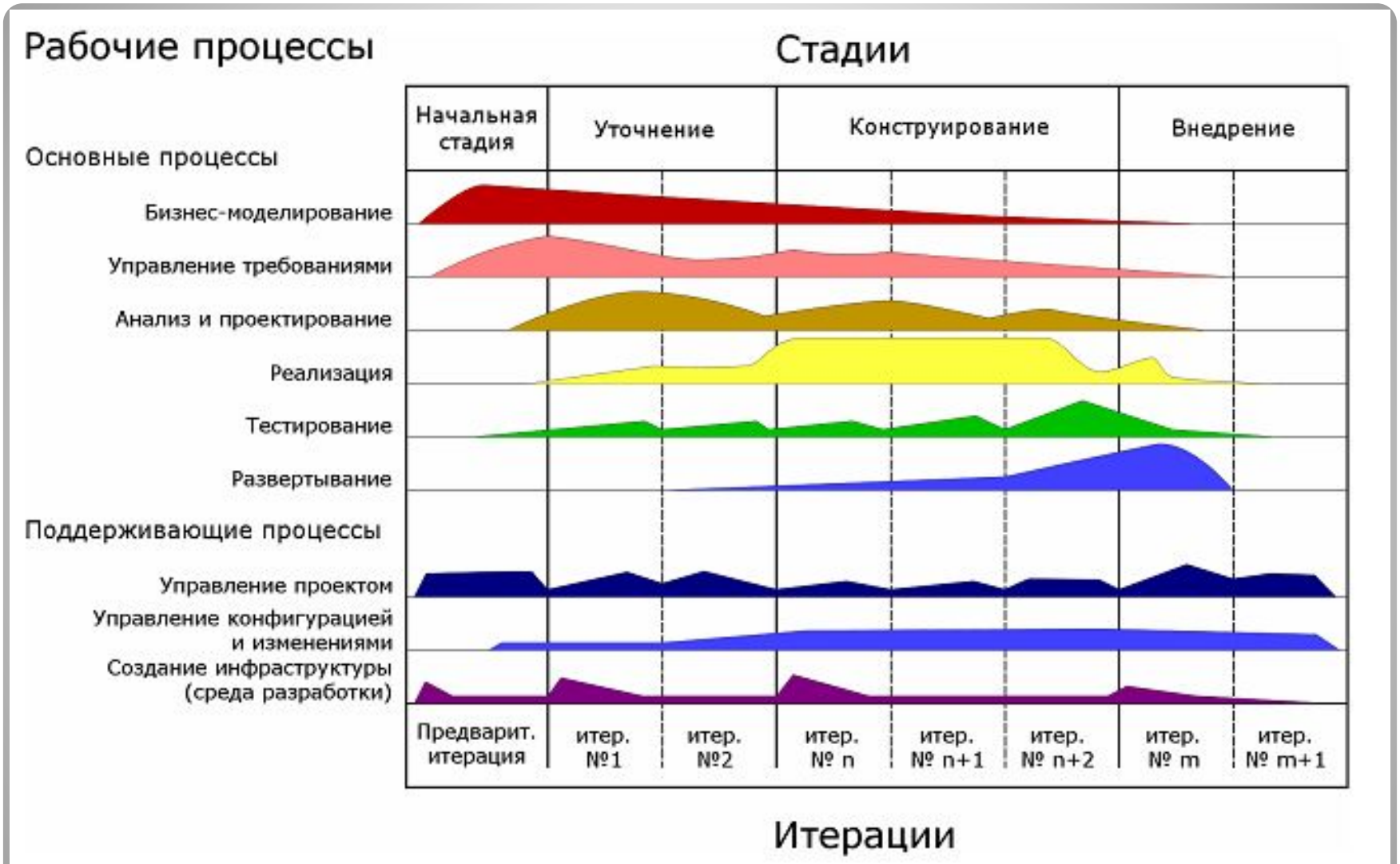
Дисциплина управления рисками

- Управление знаниями, профессиональными умениями и способностями, необходимыми для планирования, создания и сопровождения успешных решений
- Планирование процесса управления подготовкой

Дисциплина управления подготовкой

Новое в MSF 4.0

- Интеграция в среду разработки Visual Studio Team System
- Новый ролевой кластер в модели команды – Архитектура, отвечающий за проектирование решения
- Несколько профилей модели процессов
  - MSF for Agile development
  - MSF for CMMI


Рабочие процессы
Стадии
Начальная стадия
Уточнение
Конструирование
Внедрение
Основные процессы
Бизнес-моделирование
Управление требованиями
Анализ и проектирование
Реализация
Тестирование
Развертывание
Поддерживающие процессы
Управление проектом
Управление конфигурацией и изменениями
Создание инфраструктуры (среда разработки)
Предварит. итерация
итер. №1
итер. №2
итер. № n
итер. № n+1
итер. № n+2
итер. № m
итер. № m+1
Итерации

RUP

- Ранняя идентификация и непрерывное (до окончания проекта) устранение основных рисков.
- Концентрация на выполнении требований заказчиков к исполняемой программе (анализ и построение модели прецедентов (вариантов использования)).
- Ожидание изменений в требованиях, проектных решениях и реализации в процессе разработки.
- Компонентная архитектура, реализуемая и тестируемая на ранних стадиях проекта.
- Постоянное обеспечение качества на всех этапах разработки проекта (продукта).
- Работа над проектом в сплочённой команде, ключевая роль в которой принадлежит архитекторам.

RUP

- **1. Начало (Inception)**
- Формируются видение и границы проекта.
- Создается экономическое обоснование (business case).
- Определяются основные требования, ограничения и ключевая функциональность продукта.
- Создается базовая версия модели прецедентов.
- Оцениваются риски.
- При завершении начальной фазы оценивается достижение *вехи целей жизненного цикла* (англ. *Lifecycle Objective Milestone*), которое предполагает соглашение заинтересованных сторон о продолжении проекта.

RUP

- **2. Уточнение (Elaboration)**
- В фазе уточнение производится анализ предметной области и построение исполняемой архитектуры. Это включает в себя:
- Документирование требований (включая детальное описание для большинства прецедентов).
- Спроектированную, реализованную и оттестированную исполняемую архитектуру.
- Обновленное экономическое обоснование и более точные оценки сроков и стоимости.
- Сниженные основные риски.
- Успешное выполнение фазы разработки означает достижение *вехи архитектуры жизненного цикла* (англ. *Lifecycle Architecture Milestone*).

RUP

- **3. Построение (Construction)**
- Во время этой фазы происходит реализация большей части функциональности продукта. Фаза Построение завершается первым внешним релизом системы и вехой начальной функциональной готовности (Initial Operational Capability).
- **4. Внедрение (Transition)**
- Во время фазы Внедрение создается финальная версия продукта и передается от разработчика к заказчику. Это включает в себя программу бета-тестирования, обучение пользователей, а также определение качества продукта. В случае, если качество не соответствует ожиданиям пользователей или критериям, установленным в фазе Начало, фаза Внедрение повторяется снова. Выполнение всех целей означает достижение вехи готового продукта (Product Release) и завершение полного цикла разработки.

RUP

- Основные идеи:
- Личности и их взаимодействия важнее, чем процессы и инструменты;
- Работающее программное обеспечение важнее, чем полная документация;
- Сотрудничество с заказчиком важнее, чем контрактные обязательства;
- Реакция на изменения важнее, чем следование плану.

XP

- Короткий цикл обратной связи (Fine scale feedback)
  - Разработка через тестирование (Test driven development)
  - Игра в планирование (Planning game)
  - Заказчик всегда рядом (Whole team, Onsite customer)
  - Парное программирование (Pair programming)
- Непрерывный, а не пакетный процесс
  - Непрерывная интеграция (Continuous Integration)
  - Рефакторинг (Design Improvement, Refactor)
  - Частые небольшие релизы (Small Releases)
- Понимание, разделяемое всеми
  - Простота (Simple design)
  - Метафора системы (System metaphor)
  - Коллективное владение кодом (Collective code ownership) или выбранными шаблонами проектирования (Collective patterns ownership)
  - Стандарт кодирования (Coding standard or Coding conventions)
- Социальная защищенность программиста (Programmer welfare):
  - 40-часовая рабочая неделя (Sustainable pace, Forty hour week)

XP

XP

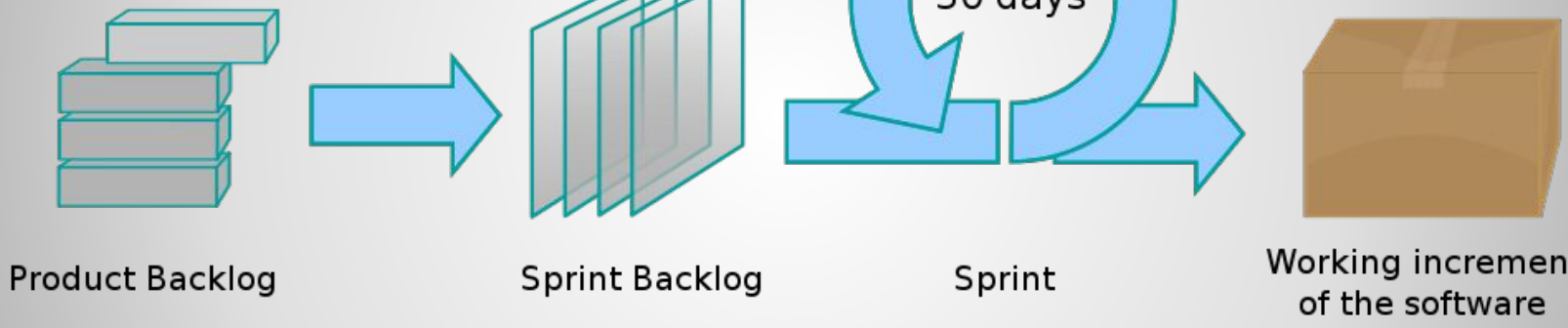
- Тестирование
  - тестирование модулей (unit testing);
  - приемочное тестирование (acceptance testing).
- Игра в планирование
  - быстро сформировать приблизительный план работы и постоянно обновлять его
  - customer stories
- Заказчик всегда рядом
  - не тот, кто оплачивает счета, а тот, кто на самом деле использует систему

- Стандарты кодирования
  - члены команды не тратят время на глупые споры о вещах, которые фактически никак не влияют на скорость работы над проектом;
  - обеспечивается эффективное выполнение остальных практик.
- Коллективное владение
  - каждый член команды несёт ответственность за весь код

XP

Lean – бережливая разработка

- **Исключение затрат**. Затратами считается всё, что не добавляет ценности для потребителя. В частности: излишняя функциональность; ожидание (паузы) в процессе разработки; нечёткие требования; бюрократизация; медленное внутреннее сообщение.
- **Акцент на обучении.** Короткие циклы разработки, раннее тестирование, частая обратная связь с заказчиком.
- **Предельно отсроченное принятие решений.** Решение следует принимать не на основе предположений и прогнозов, а после открытия существенных фактов.
- **Предельно быстрая доставка заказчику.** Короткие итерации.
- **Мотивация команды.** Нельзя рассматривать людей исключительно как ресурс. Людям нужно нечто большее, чем просто список заданий.
- **Интегрирование.** Передать целостную информацию заказчику. Стремиться к целостной архитектуре. Рефакторинг.
- **Целостное видение.** Стандартизация, установление отношений между разработчиками. Разделение разработчиками принципов бережливости. «Мыслить широко, действовать мало, промахиваться быстро; учиться стремительно».


24 h
30 days
Product Backlog
Sprint Backlog
Sprint
Working increment
of the software

Scrum

Scrum

- *ScrumMaster* — тот, кто ведёт *Scrum* митинги и следит, чтобы при этом соблюдались все принципы *Scrum*;
- *Владелец Продукта* (*Product Owner*) — человек, который представляет интересы конечных пользователей и других заинтересованных в продукте сторон;
- кросс-функциональная *Команда* (*Scrum Team*), состоящая как из разработчиков, так и из тестировщиков, архитекторов, аналитиков и т. д. (при этом размер команды в идеале составляет 7±2 человека)

Курица говорит свинье: «Давай откроем ресторан!» Свинья смотрит на курицу и отвечает: «Хорошая идея, и как мы его назовем?» Курица подумала и говорит: «Почему бы не назвать 'Яичница с беконом'?». «Так не пойдёт», — отвечает свинья, «ведь тогда мне придётся полностью посвятить себя проекту, а ты будешь вовлечена только частично.»

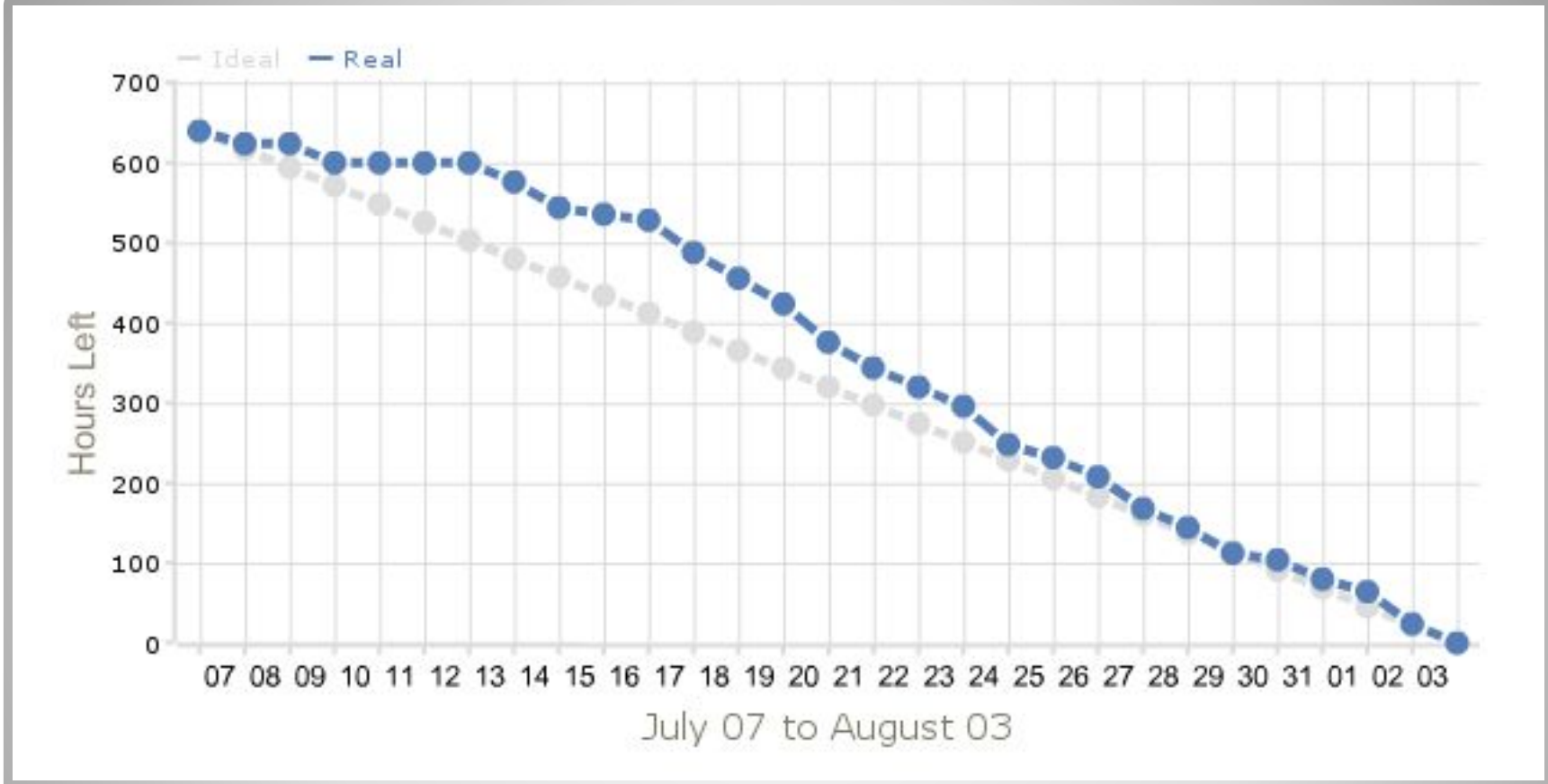
- **«Свиньи»**

  Свиньи полностью включены в проект, в Скрам процесс, они как бы едины со «своей сущностью» на производственной линии.
  - **Владелец Продукта (*Product Owner*)**
  - **Руководитель (*ScrumMaster*)**
  - **Команда (*Scrum Team*)**
- **«Цыплята»**
  - **Пользователи (*Users*)**
  - **Клиенты, Продавцы (*Stakeholders*)**
  - **Эксперты-консультанты (*Consulting Experts*)**

Scrum

- **Product backlog** — это документ, содержащий список требований к функциональности, которые упорядочены по степени важности. Product backlog представляет собой список того, что должно быть реализовано. Элементы этого списка называются «историями» (*user story*) или элементами backlog'а (*backlog items*). Product backlog открыт для редактирования для всех участников Scrum-процесса.
- **Sprint Backlog** — содержит функциональность, выбранную *Product Owner* из *Product Backlog*. Все функции разбиты по задачам, каждая из которых оценивается командой. Каждый день команда оценивает объем работы, который нужно проделать для завершения задач.
- **Burndown chart** — показывает, сколько уже исполнено и сколько ещё остаётся сделать.

Scrum


Ideal
Real
700
600
500
400
300
200
100
0
Hours Left
07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 01 02 03
July 07 to August 03

Scrum

- **Планирование спринта (Planning Meeting)**
  - Происходит в начале итерации.
  - Выбирается объём работ, обязательства по выполнению которой за спринт принимает на себя команда
  - Обсуждается и определяется, каким образом будет реализован этот объём работ
  - Каждая запись PBL принятая к реализации разбивается на подзадачи, которые оцениваются в идеальных человеко-часах
  - Ограничен 4-8 часами в зависимости от продолжительности итерации, опыта команды и т. п.

Scrum

- **Митинг (Daily Scrum)**
  - Происходит каждый день в течение спринта. Является «пульсом» хода спринта. Митингу присущи следующие ограничения:
  - начинается точно вовремя;
  - все могут наблюдать, но только «свиньи» говорят;
  - ограничен во времени 15-ю минутами;
  - проводится в одном и том же месте в течение спринта.
  - В течение митинга каждый член команды отвечает на 3 вопроса.
  - Что сделано с момента предыдущего митинга до текущего?
  - Что будет сделано с момента текущего митинга до следующего?
  - Какие проблемы мешают достижению целей спринта? (Над решением этих проблем работает *ScrumMaster*. Обычно это решение проходит за рамками митинга и в составе лиц, непосредственно затронутых данным препятствием.)

Scrum

- **Демонстрация (Demo Meeting)**
  - Происходит в конце итерации (спринта).
  - Команда демонстрирует инкремент функциональности продукта всем заинтересованным лицам.
  - Привлекается максимальное количество зрителей.
  - Все члены команды участвуют в демонстрации (один человек на демонстрацию или каждый показывает, что сделал за спринт).
  - Ограничена 4-мя часами в зависимости от продолжительности итерации и инкремента продукта.
- **Ретроспектива (Retrospective Meeting)**
  - Все члены команды рассказывают своё отношение к ходу прошедшего спринта.
  - Отвечают на два основных вопроса (Что было сделано хорошо в прошедшем спринте? Что надо улучшить или не допускать в следующем?).
  - Выполняют улучшение процесса разработки (решают вопросы и фиксируют удачные решения).
  - Ограничена 1—3-мя часами.

Scrum