Методология экстремального программирования Scrum

Кожевина Елена Игоревна

Вспоминая пройденное…

Процесс разработки ПО (жизненный цикл ПО): набор действий и связанных с ними результатов, направленных на разработку и/или развитие программного продукта.

Вспоминая пройденное…

Модель процесса разработки ПО:

не «идеальное» и «всеобъемлющее» описание процесса разработки ПО, а абстракция, описывающая различные подходы.

Гибкая методология разработки (AGILE)

Подходы с использванием итеративной модели – серии коротких циклов(2-3 нед), выглядящими как миниатюрный программный проект с этапами:

- ПЛАНИРОВАНИЕ
- АНАЛИЗ ТРЕБОВАНИЙ
- ПРОЕКТИРОВАНИЕ
- ПРОГРАММИРОВАНИЕ
- ТЕСТИРОВАНИЕ
- ДОКУМЕНТИРОВАНИЕ

Манифест гибкой методологии разработки ПО

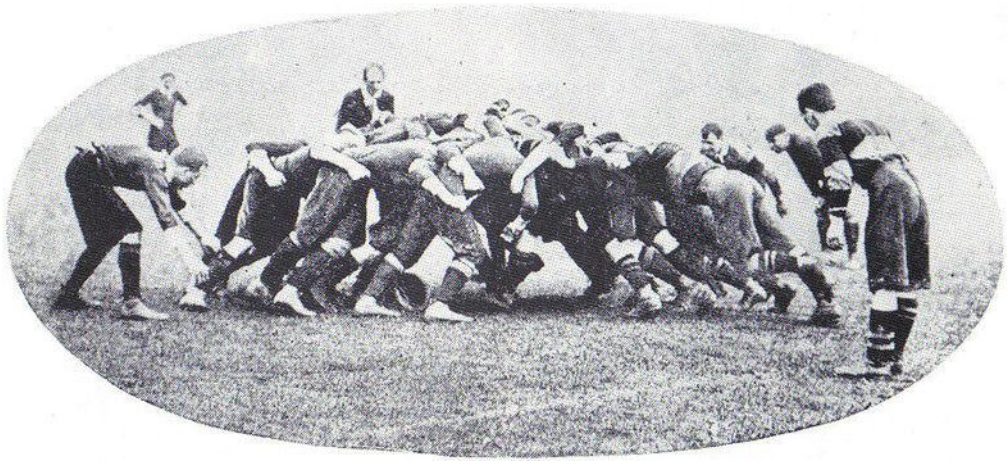
Выпущен в феврале 2001 года в штате Юта, США.

Манифест гибкой методологии разработки ПО

- Люди и взаимодействия между ними важнее процессов и средств.
- Работающие программы важнее идеальной документации.
- Сотрудничество с заказчиком важнее переговоров по условиям контракта.
- Готовность к изменениям важнее следования плану.

SCRUM

Гибкая методология управления проектами, применяющаяся при разработке программ



Создатели SCRUM

Майкл Бидл

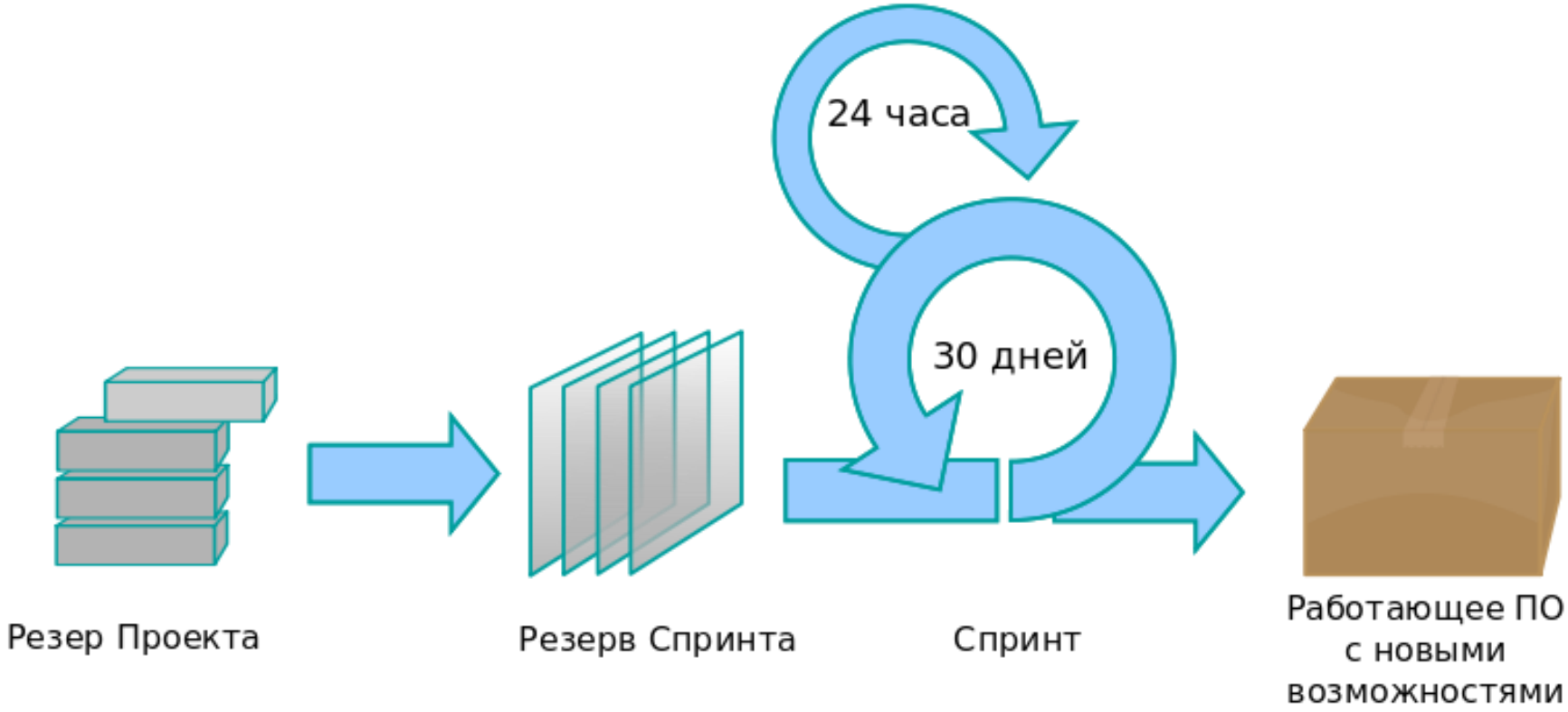
Кен Швабер

Agile Software Development with SCRUM, 2001

SCRUM

Набор принципов, на которых стоятся небольшие по времени итерации(спринты), в конце которых пользователю представляется рабочее ПО с возможностями, для которых определен наибольший приоритет

SCRUM


24 часа
30 дней
Резер Проекта
Резерв Спринта
Спринт
Работающее ПО
с новыми
возможностями

Спринт

Итерация, в течении которой создается функциональный рост ПО, жестко фиксирован по времени(2-4 нед.).

Какой лучше по длине?

Product backlog

Список требований, упорядоченный по важности для реализации. Требования называются user story («истории пользователя»).

Product backlog

Product backlog (пример)					
ID	Название	Важность	Предварительная оценка	Как продемонстрировать	Примечания
1	Депозит	30	5	Войти в систему, открыть страницу депозита, положить на счет €10, перейти на страницу баланса и проверить, что он увеличился на €10.	Нужна UML диаграмма последовательности. Пока что не стоит беспокоиться про шифрование данных.
2	Просмотр журнала личных транзакций	10	8	Войти в систему; перейти на страницу транзакций; положить деньги на счет; вернуться на страницу транзакций; проверить, что новая транзакция появилась в списке.	Чтобы избежать больших запросов к базе данных, стоит воспользоваться постраничным выводом информации. Дизайн такой же, как и у страницы просмотра пользователей.

User story

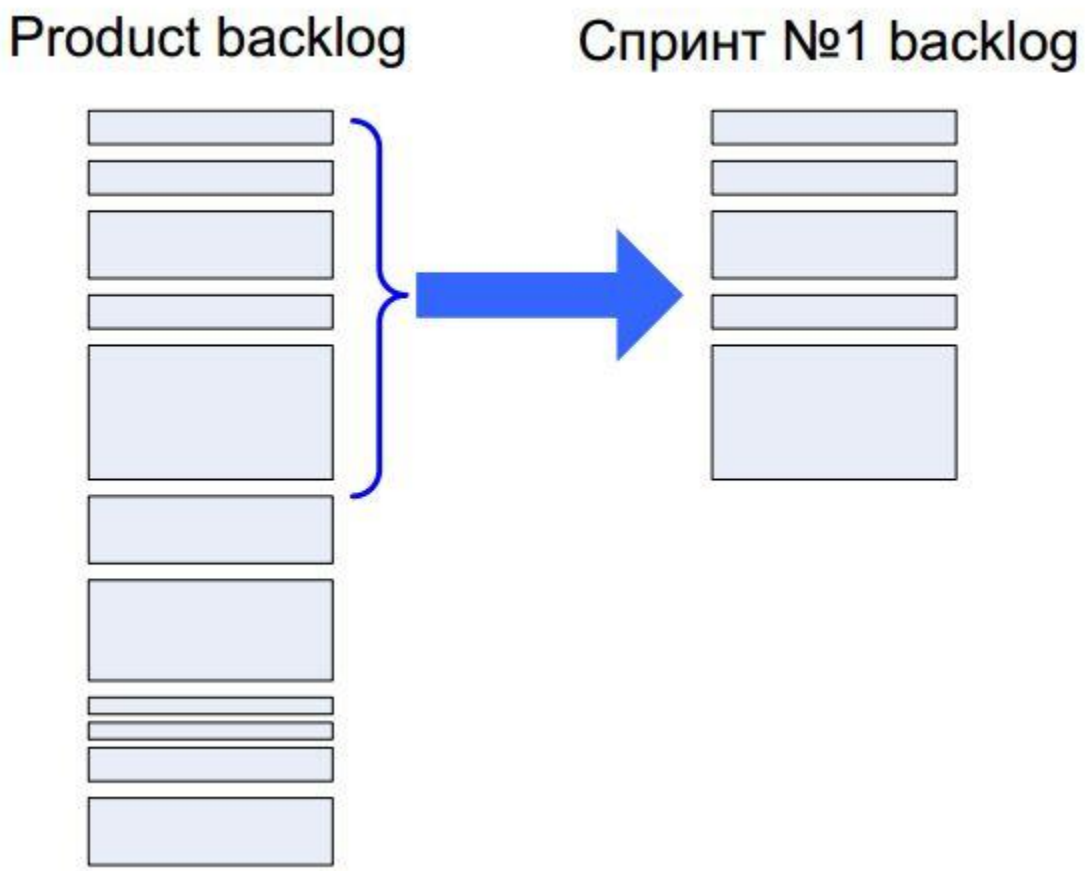
- ID
- Название (описание).
- Важность (мнение владельца).
- Предварительная оценка (часы).
- Как продемонстрировать (тест).
- Примечания.

Как истории попадают в спринт?

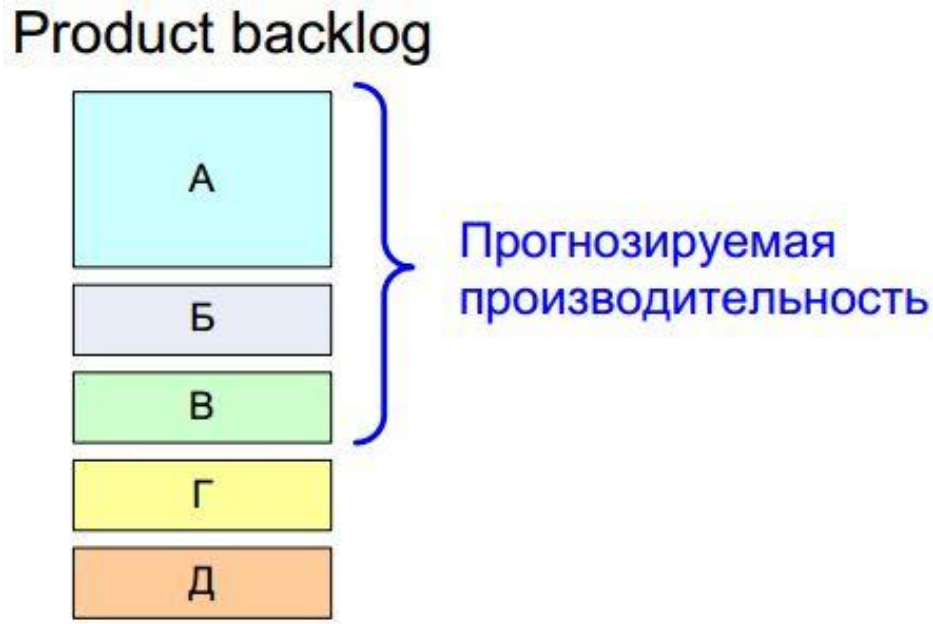
Команда решает, какие задачи попадут в спринт на основе оценки производительности и интуиции.

Но product owner может влиять на решение.

Sprint №1 backlog

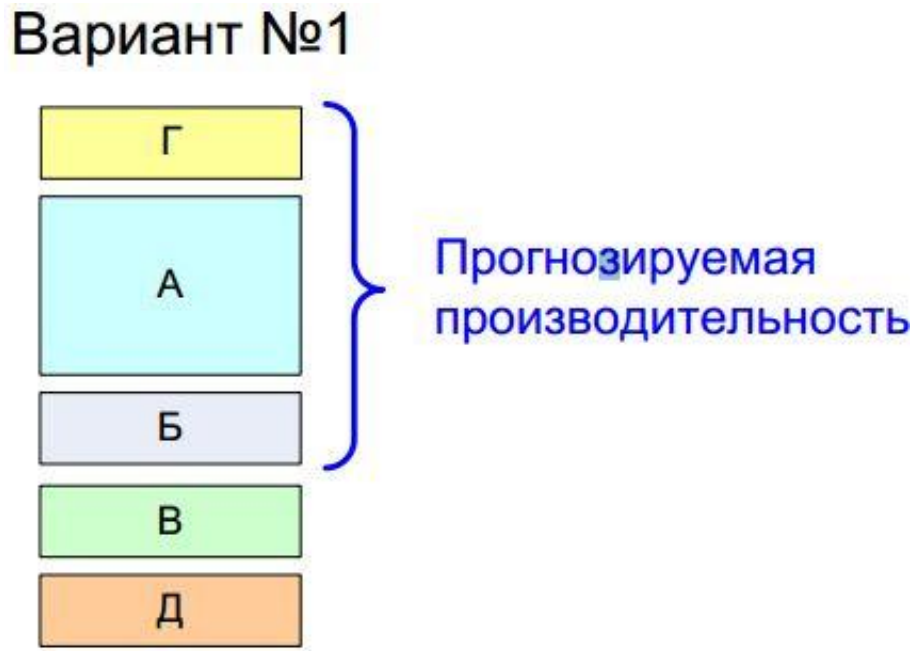


Ситуация при планировании спринта



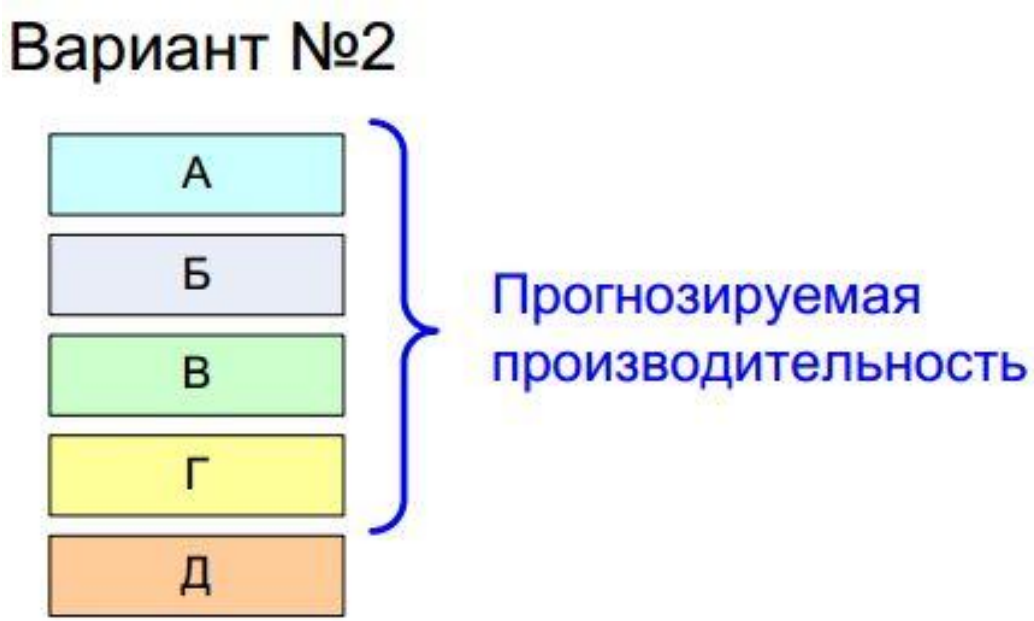
Product owner: «Хочу, чтобы история Г попала в спринт»

Ситуация при планировании спринта



Вариант №1 – изменение приоритетов истории Г.

Ситуация при планировании спринта



Вариант №3 – изменение объемов истории А.

Ситуация при планировании спринта

Вариант №3 – разбиении истории А на подзадачи.

Решение команды – подсчет производительности

Число в прямоугольнике – оценка сложности истории.

Решение команды – подсчет производительности

Ценность незаконченного спринта – нулевая.
Мы должны иметь готовый продукт.

Решение команды – подсчет производительности

Определяем доступные ресурсы.

Но максимально продуктивные дни бывают редко.

Решение команды – подсчет производительности

Прогнозируемая производительность этого спринта

(доступные человеко-дни) x (фокус-фактор) = (прогнозируемая производительность)

Вводим фокус-фактор – коэффициент сосредоточенности команды на работе.

Решение команды – подсчет производительности

Фокус-фактор последнего спринта

$$(\text{фокус-фактор}) = \frac{(\text{действительная производительность})}{(\text{доступные человеко-дни})}$$

Высчитываем фокус-фактор по результатам последнего спринта.

Решение команды – подсчет производительности

За последний спринт реализовано задач в сумме на 18 story-point'ов за 45 дней.

Теперь наш прогноз – 20 story point'ов в этом спринте.

Решение команды – подсчет производительности

Включаем истории примерно на 20 story point'ов. При сомнениях лучше выбрать меньше историй.

Встреча – планирование спринта №1

Посвящена историям из backlog’а: их оценке, расстановке важности, уточнению требований , разбиению на части. При этом используются *учетные карточки*, прикрепленные на стену или стол.

Встреча – планирование спринта №1

Пример учетной карточки.

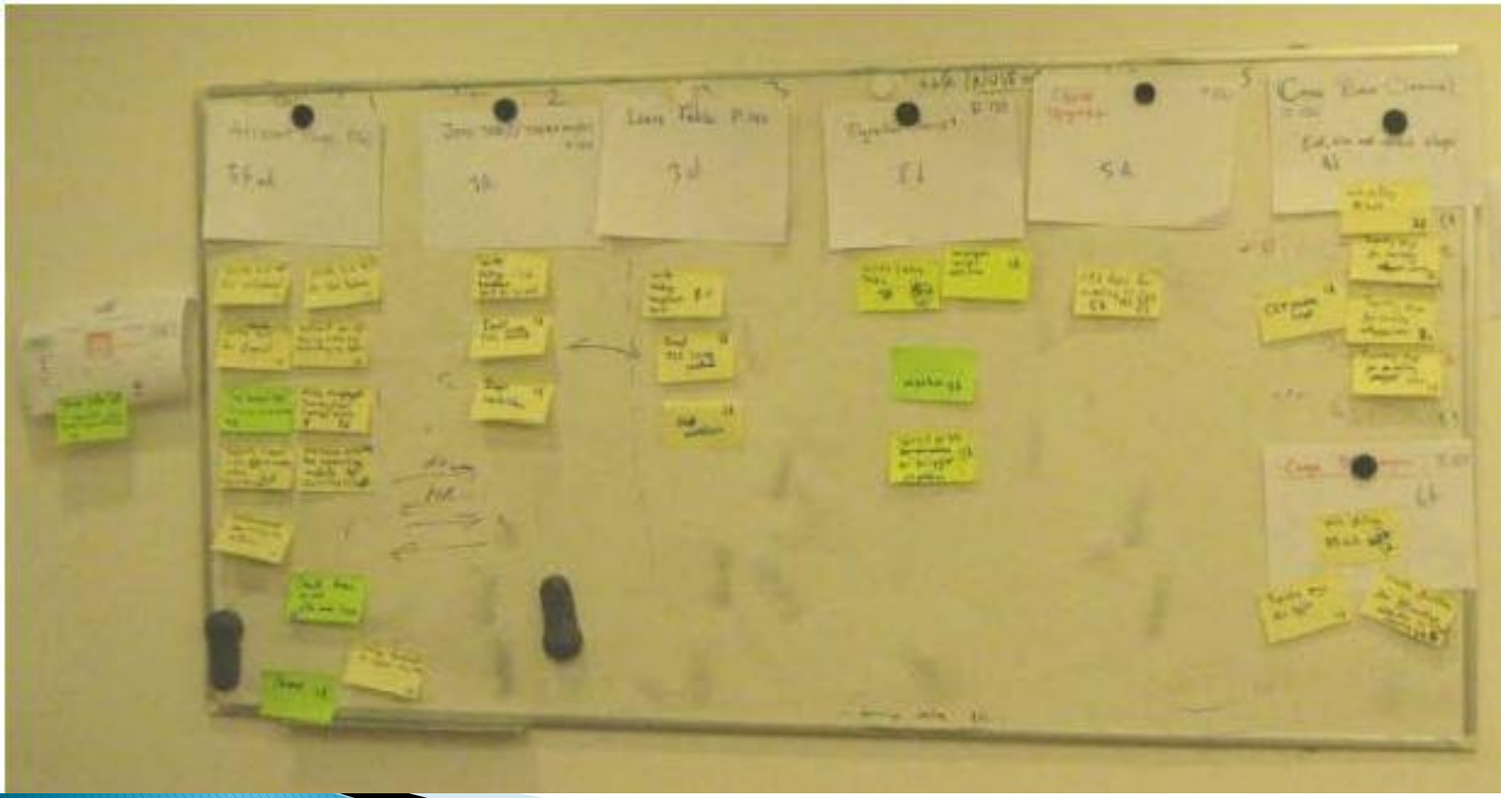
Спринт №1


Более важно
Менее важно
Депозит
Автомат. обновление
Админка: вход
Админка: Управление пользователями
Снятие со счёта
Тест произво-ти
Шифрование паролей

Спринт №1


Более важно
Менее важно
9д
Депозит
Приёмочные тесты 2д
DAO 3д
Дизайн БД 1д
Интеграционное тест-ие 2д
Рефакторинг 1д
14д
Автомат. обновление
Приёмочные тесты 2д
Покопаться в Tapesty 2д
Спец. для GUI 2д
Написать скрипт 8д
5д
Админка: вход
Приёмочные тесты 2д
Разраб. GUI 1д
Интеграция с JBoss 2д
12д
Админка: Управление пользователями
Приёмочные тесты 3д
Уточнить требования 2д
Дизайн для GUI (CSS) 1д
Разраб. GUI 6д
15д
Снятие со счёта
12д
Тест произво-ти
Шифрование паролей

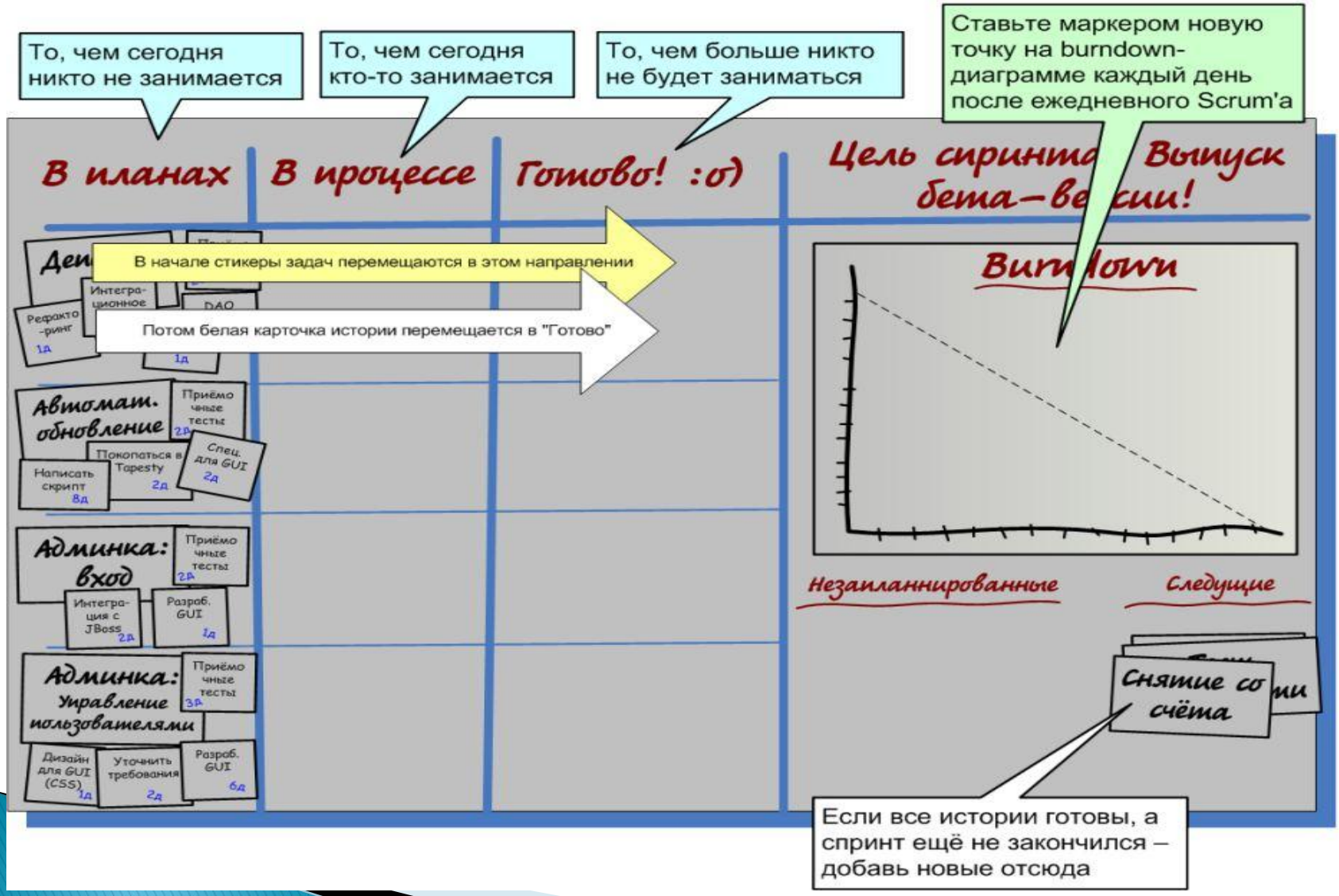
Спринт №1



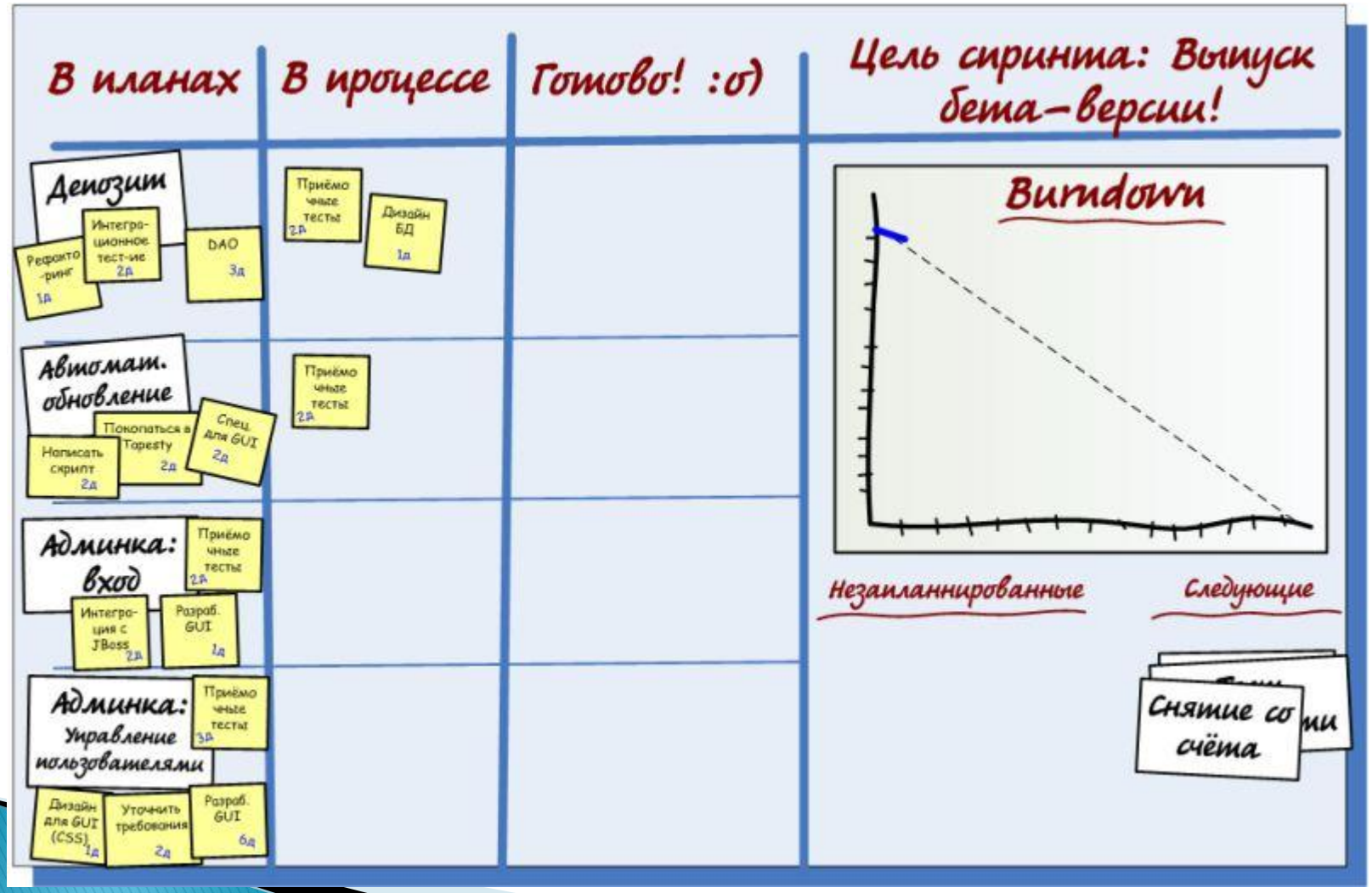
Спланировали. Что дальше?

- Планирование работы с помощью доски задач.
- Проведение ежедневного скрама в уголке обсуждений.
- Организация работы команды.

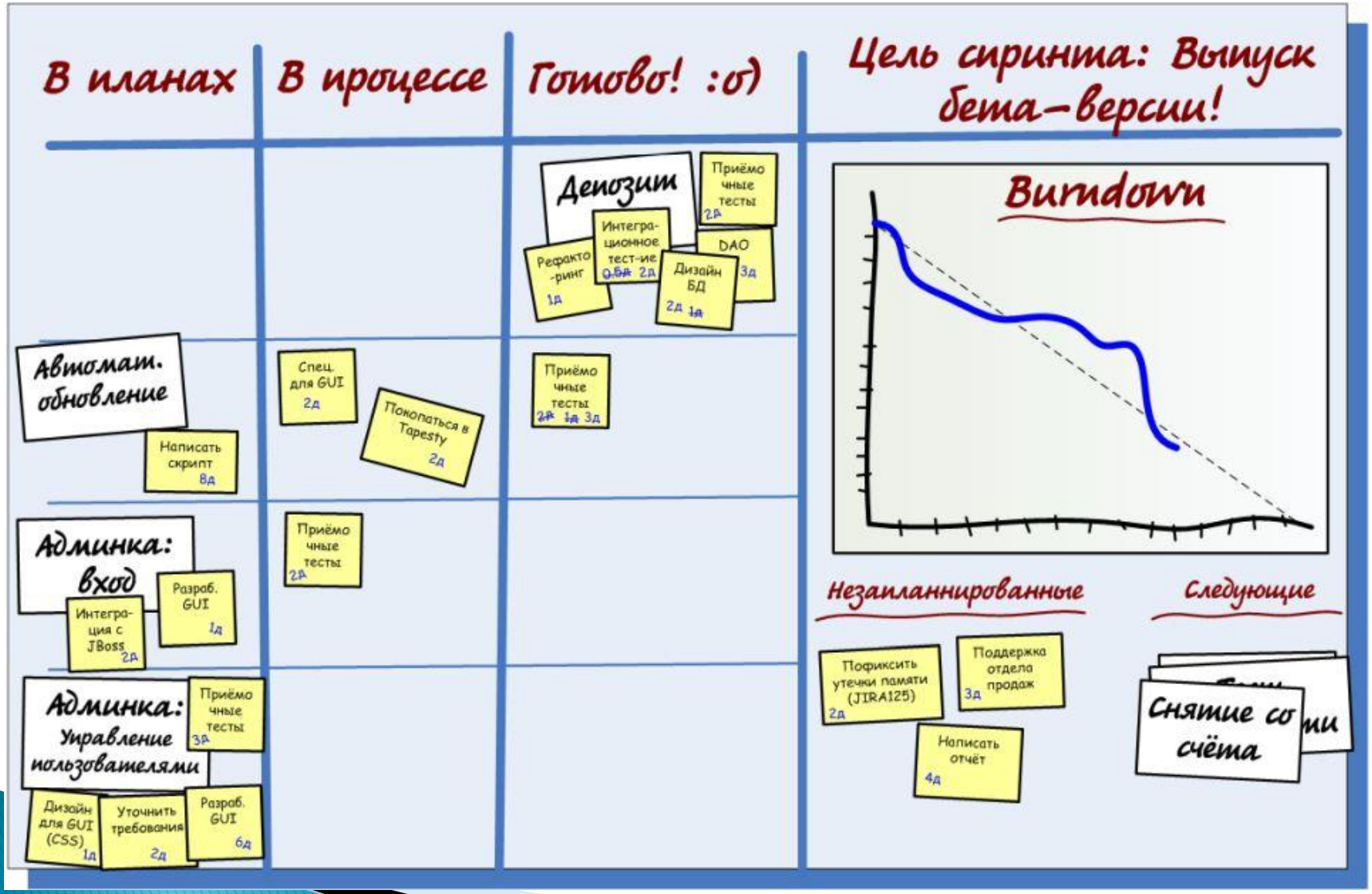
Создаем доску задач



Доска задач через день

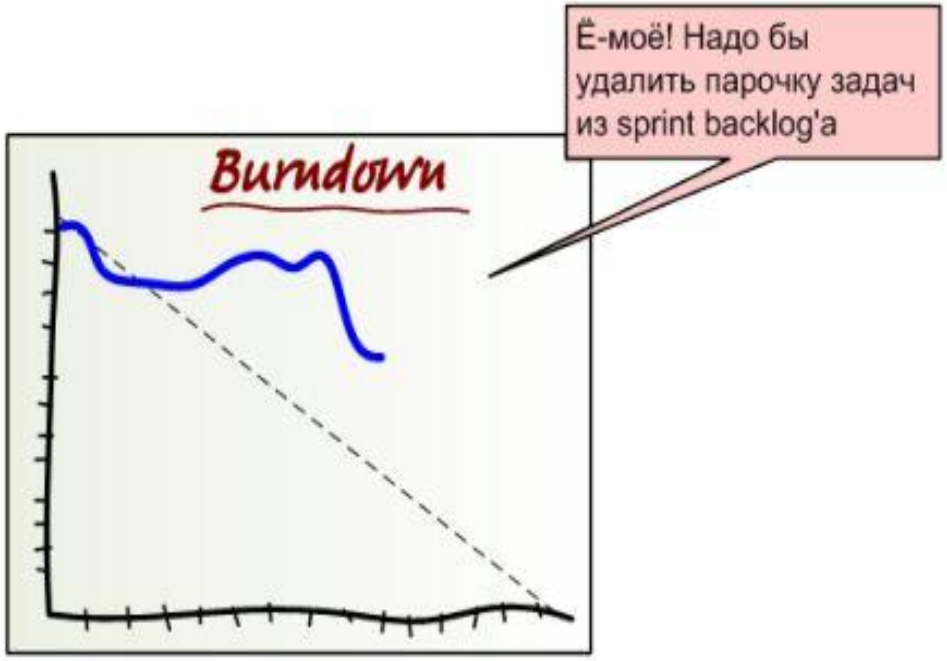


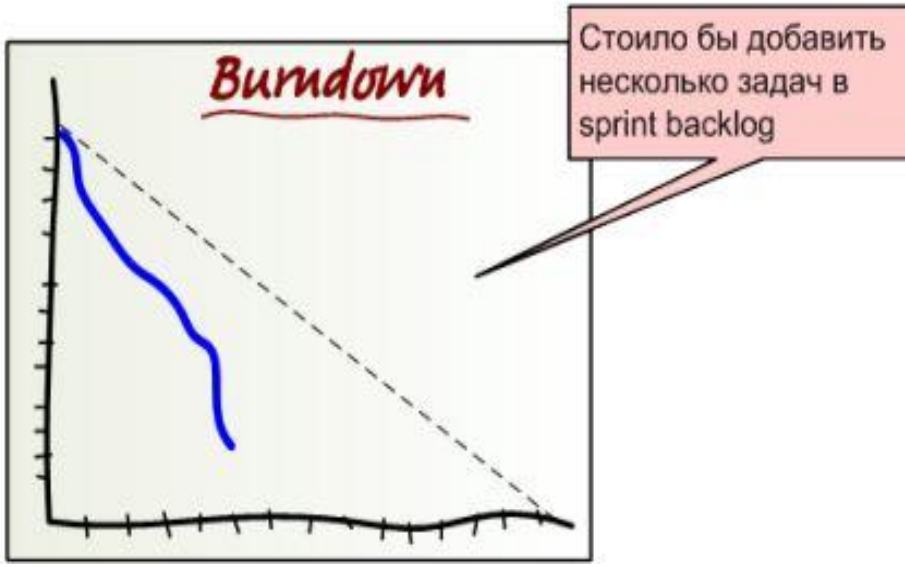
Доска задач в конце спринта



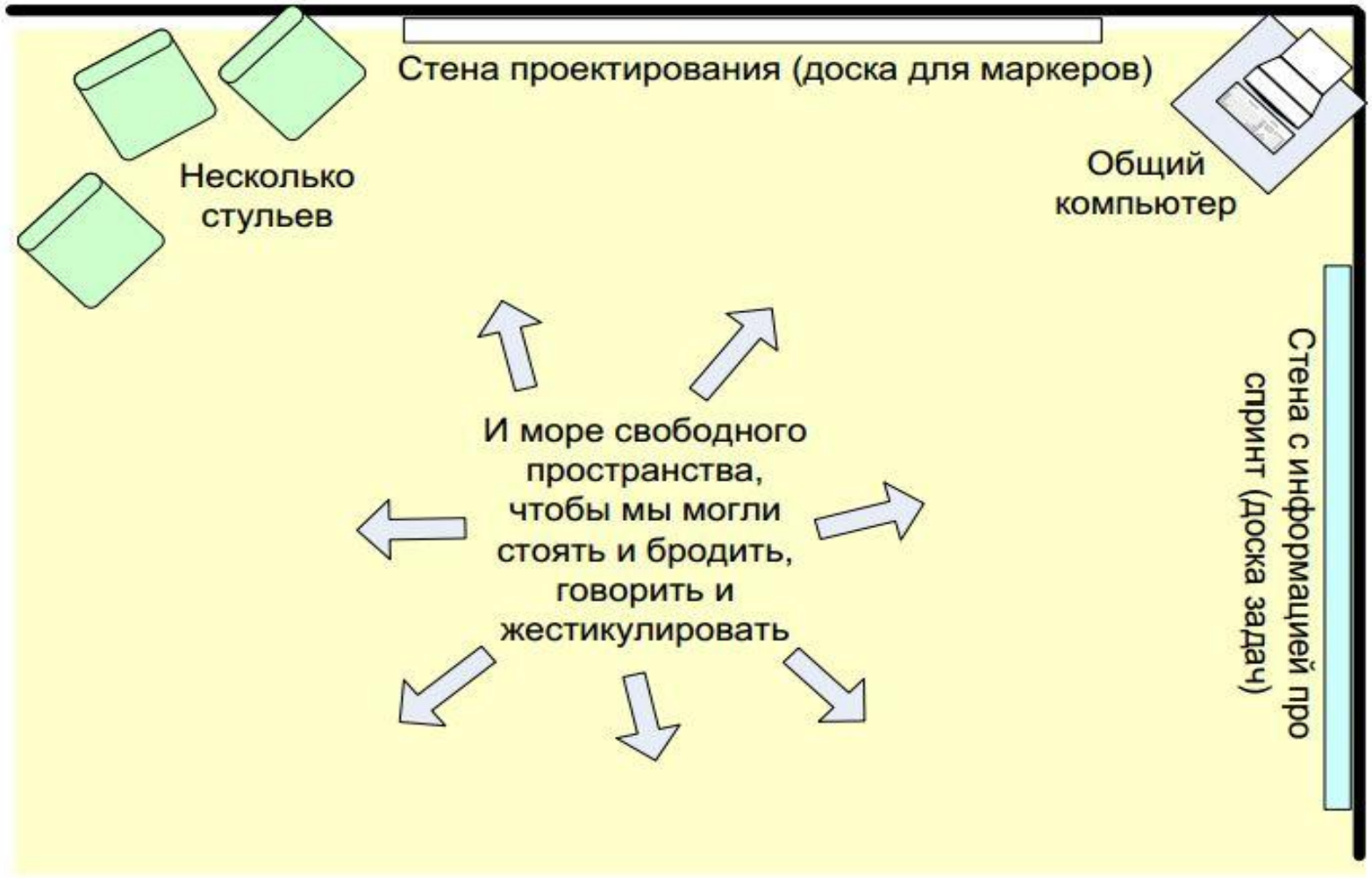
Реальная доска задач

Анализ burndown-диаграммы





Уголок обсуждений



Проведение настоящего скрама

Как организовать работу команды

- Усадить всех в пределах слышимости, близости.
- Использовать веб-камеры для удаленных сотрудников.
- Не подпускать посторонних близко.

Закончили. Что дальше?

- Проведение демонстрации для насущной критики, для воодушевления команды или ее «подпинывания».
- Проведение ретроспективы с участием product owner’a, scrum master’a и всей команды для учета сильных и слабых мест, анализа процесса работы и проведения изменений.

Ретроспектива в жизни

Закончили. Что дальше?

- Дать команде как можно больше отдыха – спринты изматывают. В идеале это может быть 3–4 дня.
- Начать новый спринт.

Экстремальное программирование (XP)

Extreme programming, XP.

Одна из гибких методологий разработки ПО.

История разработки этой методологии началась в первой половине 90-х годов.

Создатель XP – Кент Бек

Положения XP

- Короткий цикл обратной связи.
- Непрерывный, а не пакетный процесс.
- Понимание, разделяемое всеми.
- Социальная защищенность программиста.

Короткий цикл обратной связи

- Разработка через тестирование.
- Игра в планирование.
- Заказчик всегда рядом.
- Парное программирование.

Непрерывный процесс

- Непрерывная интеграция.
- Рефакторинг.
- Частые небольшие релизы.

Понимание

- Простота.
- Метафора системы.
- Коллективное владение кодом.
- Стандарт кодирования.

Социальная защищенность

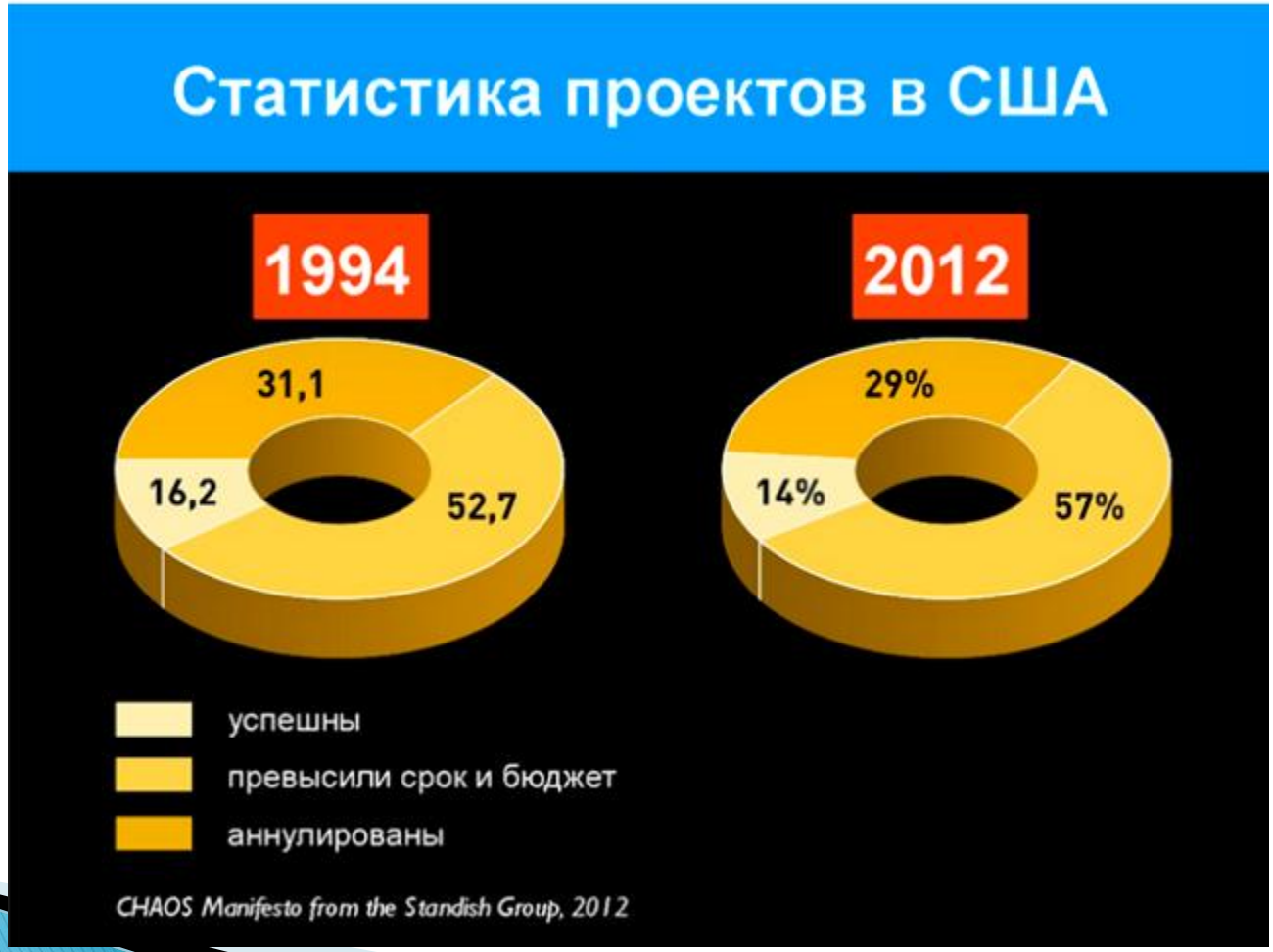
- 40-часовая рабочая неделя.

Сочетание XP и SCRUM

Хэнриг Книберг, автор книги «Scrum и XP: заметки с передовой», считает, что методологии Scrum могут быть успешно объединены. Они имеют много общего, однако Scrum решает вопросы управления, а XP – вопросы инженерных практик.

В своей работе Хэнриг использует большинство практик XP.

Статистика


Статистика проектов в США
1994
31,1
16,2
52,7
2012
29%
14%
57%
успешны
превысили срок и бюджет
аннулированы
CHAOS Manifesto from the Standish Group, 2012

Статистика


Статистика проектов в США
WATERFALL
29%
14%
57%
SCRUM
9%
42%
49%
успешны
превысили срок и бюджет
аннулированы
CHAOS Manifesto from the Standish Group, 2012

Что же лучше?