

Гибкие методологии разработки ПО

Часть 2

Ксения Шипина
Системный аналитик



Что изучим?

- 1 Scrum
- 2 eXtreme programming
- 3 Kanban
- 4 Scrum vs Kanban



Scrum

1



Scrum: история

1986

В Harvard Business Review вышла статья Хиротакки Такеючи и Икуджио Нонаки «Разработка нового продукта. Новые правила игры»

1993

Scrum впервые применён на практике в Easel Corporation Джеффом Сазерлендом, Джоном Скамниотлесом и Джеффом МакКенной

1995

Джефф Сазерленд и Кен Швабер публикуют статью «Scrum. Процесс разработки»

1

2

3



Scrum: история

2002

Кен Швабер, Майк Кон
и Эстер Дерби основали
Scrum Alliance

2001

Кен Швабер и Майкл Бидл
выпустили первую книгу
о Scrum «Гибкая разработка
ПО по Scrum»

2001

17 независимых
разработчиков
приняли Agile-
манифест

6

5

4



Scrum: история

2006

Джефф Сазерленд основал
собственную компанию
Scrum.inc

2009

Кен Швабер основывает
Scrum.org

2010

Опубликовано первое
руководство по Scrum.
Основные обновления
выходили в 2011, 2013,
2016 и 2020 годах

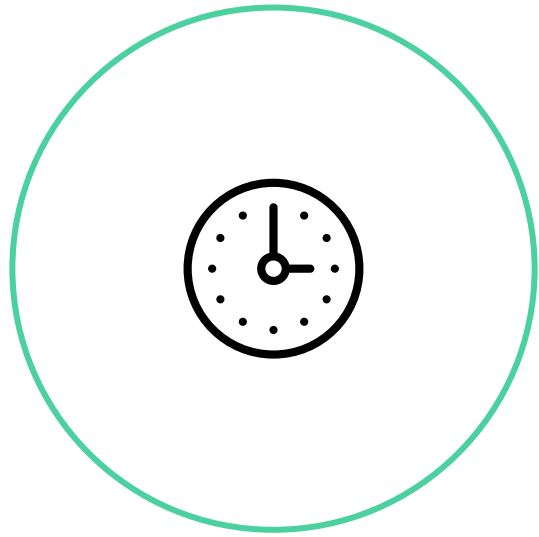
7

8

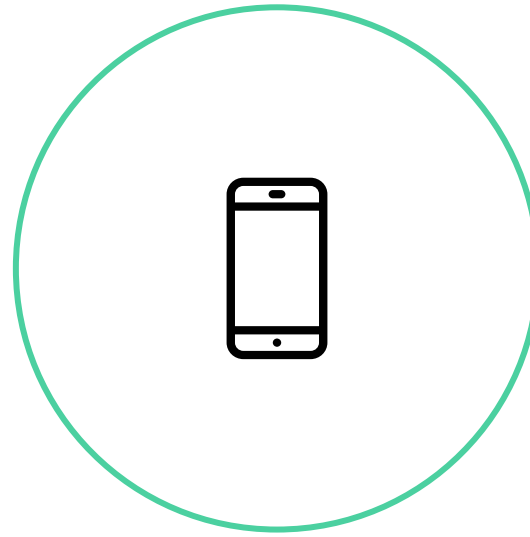
9



Scrum позволяет



За фиксированный
небольшой промежуток
времени (sprint)



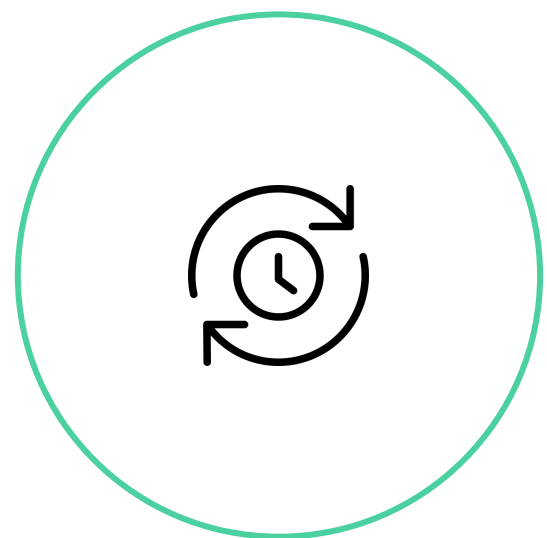
предоставить
конечному
пользователю
работающий продукт



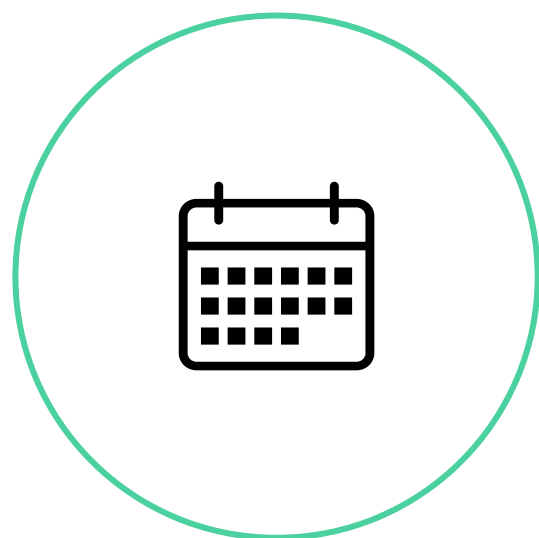
С НОВЫМИ
ВОЗМОЖНОСТЯМИ,
ПОКРЫВАЮЩИМИ
ПОТРЕБНОСТЬ
ПОЛЬЗОВАТЕЛЕЙ



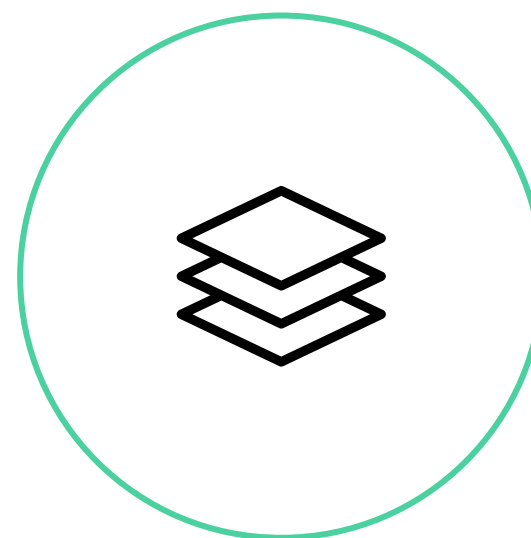
Sprint



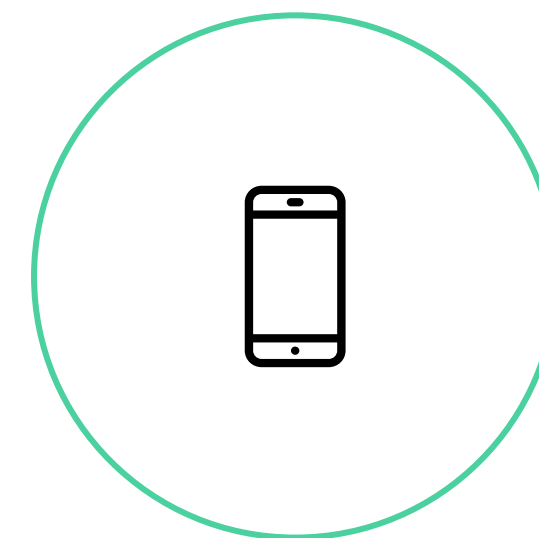
Длительность —
от 1 до 4 недель



Жёстко
фиксирован
во времени
(есть дата начала
и дата окончания)



Состоит
из обозримого
конечного
набора задач



Цель —
реализовать новую
версию
работающего
продукта и
доставить её
пользователю



Роли

Основные

- **Scrum-мастер**
следит за соблюдением всех принципов Scrum
- **Product Owner**
представляет интересы конечных пользователей
- **Команда разработки**
работает над проектом, состоит из специалистов разных профилей

.....

Дополнительные

- **Пользователи**
конечные пользователи продукта
- **Стейкхолдеры**
бизнес-заказчики, владельцы требований к продукту



Пользовательская история —

краткая формулировка намерения, описывающая
функциональность, которую система должна предоставлять
определённому
типу пользователей



Артефакты: пользовательская история (user story)

- Каждая история — это ценность для пользователя
- Завершённость каждой пользовательской истории определяется по критериям готовности задачи (definition of done)
- Каждая пользовательская история должна быть оценена командой разработки



Project backlog —

приоритизированный список всех
требований к продукту

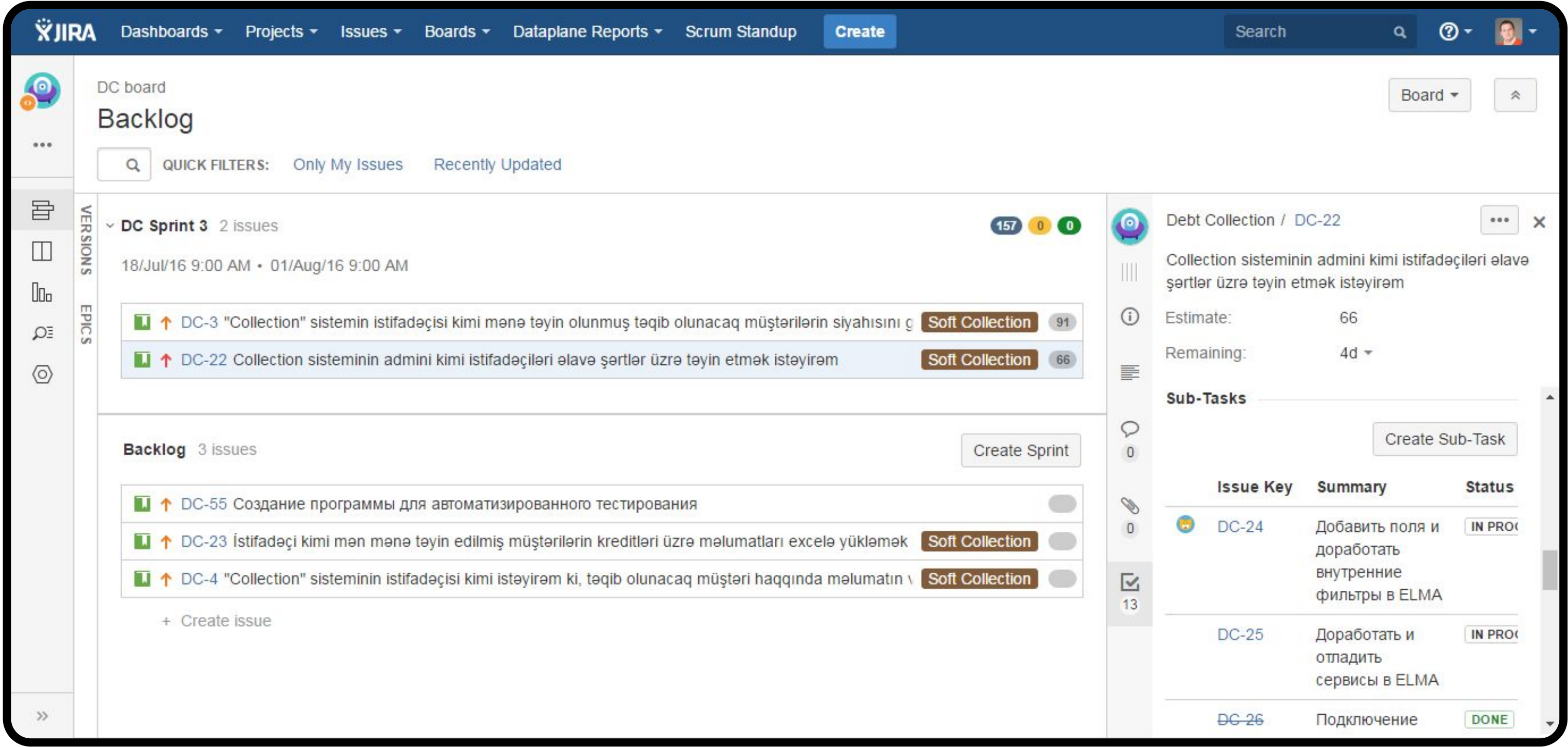


Артефакты: бэклог продукта (project backlog)

- Элементы бэклога — пользовательские истории
- Ведётся владельцем продукта
- Элементы бэклога могут быть добавлены любым участником команды разработки



Артефакты: бэклог продукта (project backlog)



Sprint backlog —

приоритизированный список требований
к продукту, которые должны быть реализованы
в конкретном спринте



Артефакты: бэклог спринта (sprint backlog)

- Выбираются из бэклога продукта владельцем продукта по приоритетности
- Количество пользовательских историй в спринте ограничивается объёмом, который может реализовать команда



Артефакты: цель спринта

Цель спринта — краткое описание бизнес-цели, которая должна быть достигнута по результатам спринта.

- Выбирается владельцем продукта
- Необходима для принятия качественных решений в процессе работы
- Обычно отражается на доске спринта



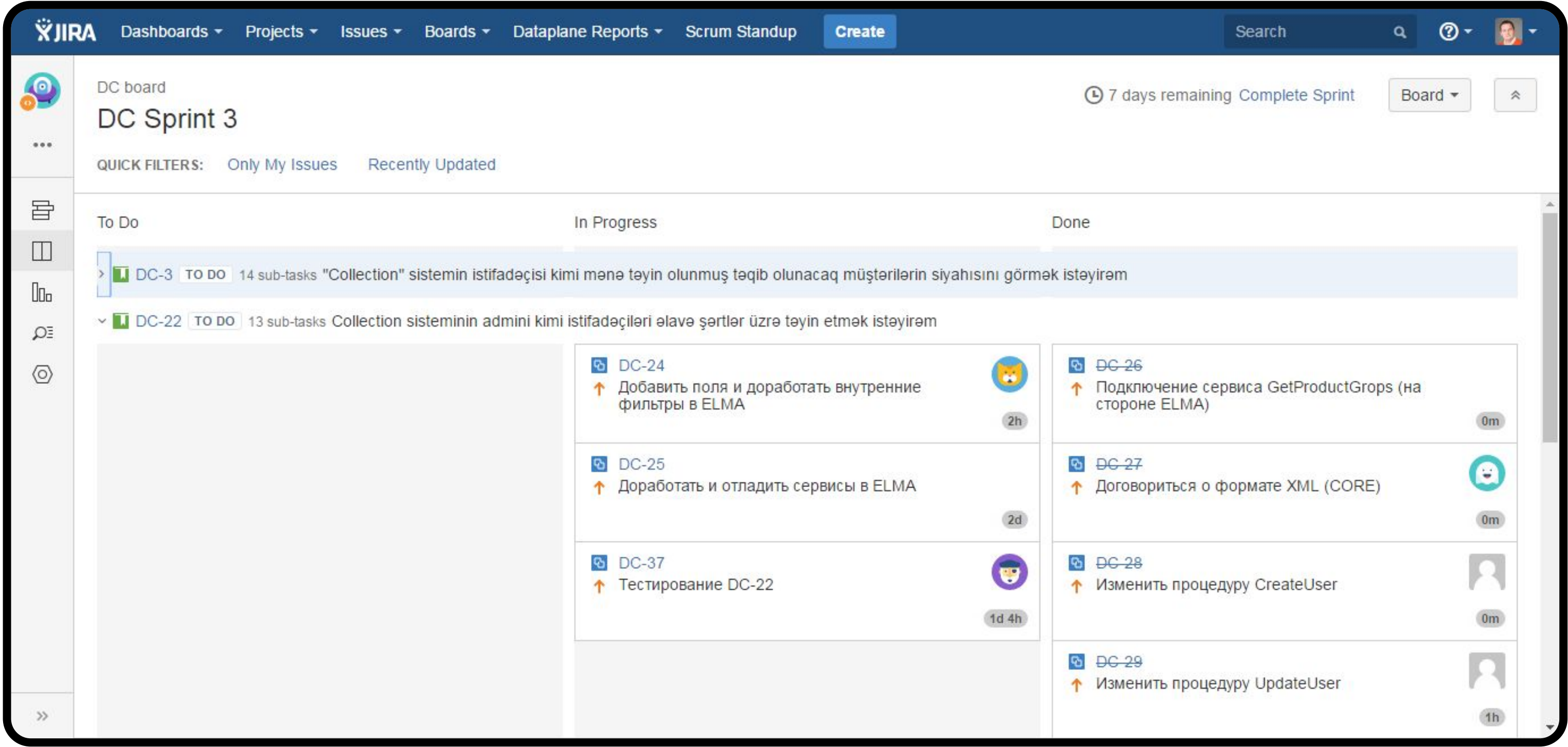
Артефакты: доска спринта

Доска спринта — инструмент визуализации потока работ в спринте.

- Состоит минимум из 3 колонок: «сделать», «в процессе», «готово»
- При запуске спринта в колонке «сделать» размещается весь бэклог спринта в порядке приоритета (сверху вниз)
- Необходима для синхронизации команды в процессе спринта, определения рисков, распределения работ
- Для визуализации доски используются инструменты Atlassian Jira, Trello, TFS



Артефакты: доска спринта



Grooming —

встреча команды разработки
и владельца продукта



Ритуалы: груминг задач (grooming)

Цели груминга:

- добавить, убрать, объединить или декомпозировать элементы бэклога на элементы-задачи уровня программирования
- уточнить или дать новые оценки
- изменить порядок следования элементов бэклога продукта
- обсудить и прояснить требования



Story points —

подход к оценке усилий, которые необходимо
приложить, чтобы реализовать задачу



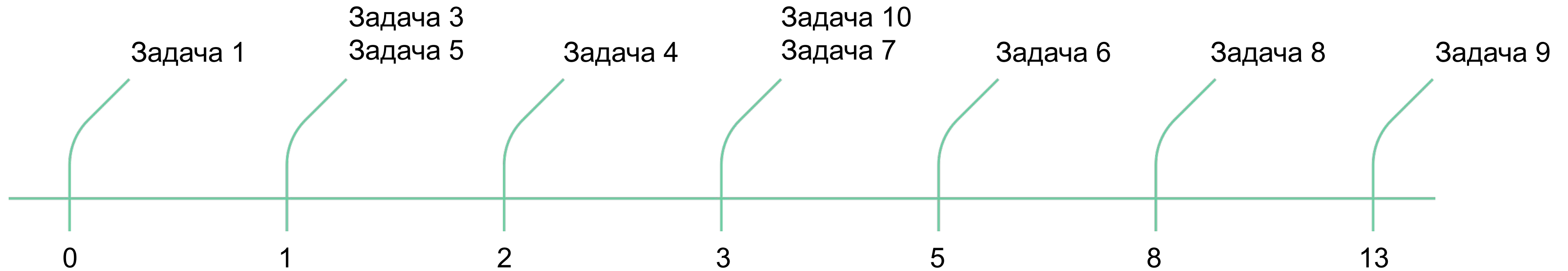
Ритуалы: оценка задач

В оценке учитывают:

- объём требуемой работы
- техническую сложность задачи
- возможные риски и неопределённость в требованиях



Ритуалы: оценка задач



Шкала оценки в Story points



Planning —

**встреча команды разработки и владельца
продукта с целью получить ответы
на следующие вопросы:**

Какова цель этого спринта или что можно
сделать за этот спринт?

Как именно будет достигнута цель спринта,
чтобы получить новую версию
работающего продукта?



Ритуалы: планирование спринта (planning)

- Проводится в начале каждого спринта
- невыполненные в прошлом спринте истории автоматически переносятся в новый спринт после уточнения оценки



Daily scrum —

ежедневная короткая встреча
команды разработки,
которая помогает понять,
как продвигается работа
по реализации
цели спринта



Ритуалы: ежедневные совещания (daily scrum)

Участники встречи по очереди отвечают на ряд вопросов:

- Что я делал вчера?
- Что я планирую сделать сегодня?
- Есть ли у меня какие-то проблемы/трудности, мешающие выполнить работу?



Sprint review —

неформальная встреча-демонстрация
разработанной за спринт функциональности, предназначенная
для получения обратной
связи по продукту, в которой
участвуют команда разработки
и все заинтересованные
лица



Ритуалы: демо (sprint review)

Демо продукта обычно проходит по следующему плану:

- Владелец продукта называет цели спринта и показывает, какие элементы бэклога были выполнены, а какие нет
- Команда разработчиков обсуждает, что прошло хорошо, а с чем возникли трудности, и как они были решены
- Команда разработчиков демонстрирует версию продукта, которая была получена в результате выполненных в спринте задач
- Команда отвечает на вопросы заинтересованных лиц и стейкхолдеров



Retrospective —

неформальная встреча всей команды продукта
с целью проанализировать прошедший спринт
и сформировать план улучшений
процесса реализации проекта
для следующего спринта



Ритуалы: ретро (retrospective)

Цель: проанализировать проблемы спринта, выявить наиболее эффективные решения и сформировать план их усовершенствования.

Проводится после обзора итогов спринта до планирования следующего спринта и должно занимать не более 3 часов.



Capacity —

объём работ, который может
выполнить команда
за один спринт



Ритуалы: расчет capacity

Динамическая величина, которая пересчитывается по результатам каждого спринта.

- Закладывается % недоступности разработчика на время отпуска/больничного
- Высчитывается средний объём работ разработчиков направления в день за 3 последних спринта и умножается на количество дней спринта и количество разработчиков направления = capacity команды направления



Пример расчёта capacity

	A	B	C	F	G
1					
2	кол-во дней в спринте	10			
3	кол-во SP в одном дне на 1 человека	2			
4	% на коммуникации	30%			
5					
6					
7					
8		kpd	сколько может реализовать SP	кол-во SP с учетом % на коммуникации	
9					
10	backend				
11	Разработчик 1	0,6	12	=B2*B3*B11	
12	Разработчик 2	0,9	18		
13	Разработчик 3	0,8	16		
14	Разработчик 4	0,6	12		
15			58	58,0	=C15-D15*B4
16	frontend				
17	Разработчик 1	0,9	18		
18			18	12,6	=C18-C18*B4
19					
20					
21					
22					
23					
24	capacity команды, SP			70,6	=F15+F18



Экстремально программирование

2



eXtreme programming



- Работал над проектом системы для расчёта зарплатных ведомостей в компании Chrysler
- Разработал методологии XP и TDD и написал об этом книгу в 1999 г.

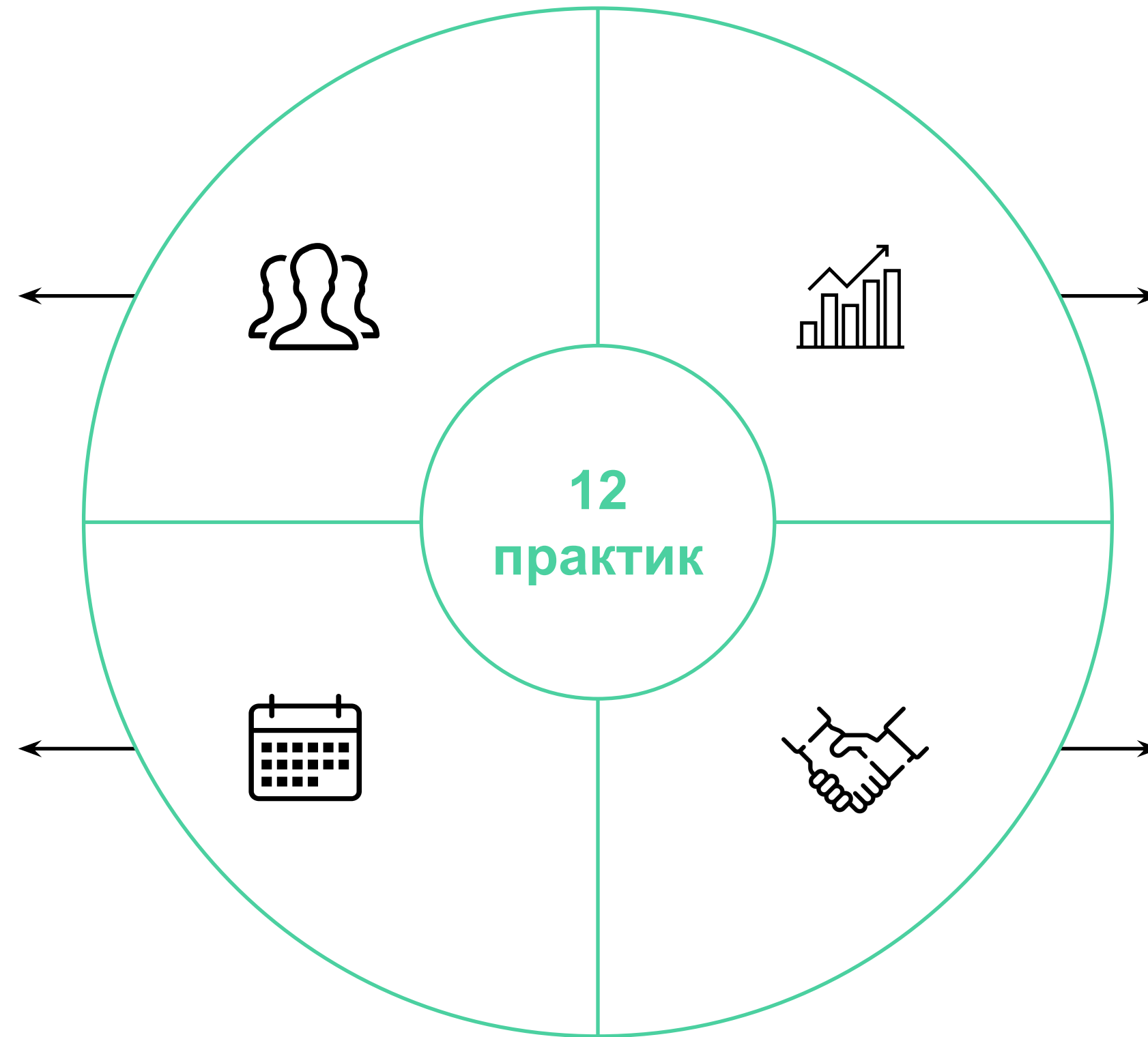
Кент Бек



eXtreme programming

Сильная
самоорганизованная
команда
(до 10 человек)

Короткие итерации
2–4 недели



Сокращение сроков
и повышение качества
разработки

Повышение доверия
заказчика

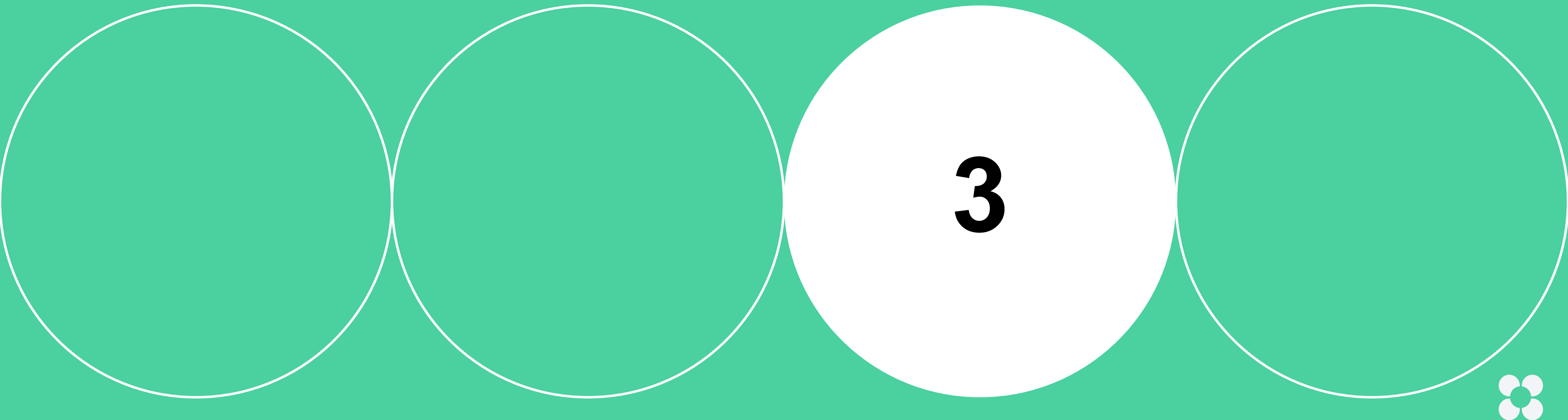


12 практик XP

- 1 Планирование процесса
- 2 Тесное взаимодействие с заказчиком
- 3 Общесистемные правила именования
- 4 Простая архитектура
- 5 Рефакторинг
- 6 Парное программирование
- 7 40-часовая рабочая неделя
- 8 Коллективное владение кодом
- 9 Единые стандарты кодирования
- 10 Небольшие релизы
- 11 Непрерывная интеграция
- 12 Тестирование



Kanban



Kanban: история



Выделил 7 видов потерь на производстве, внедрил систему организации производства и снабжения, позволяющую реализовать принцип «точно в срок»

Тайити Оно



Адаптировали философию бережливого производства к процессам разработки ПО

Мэри и Том Поппендик



Канбан: в чём суть?

Канбан — наглядная система разработки, показывающая, что необходимо производить, когда и сколько.

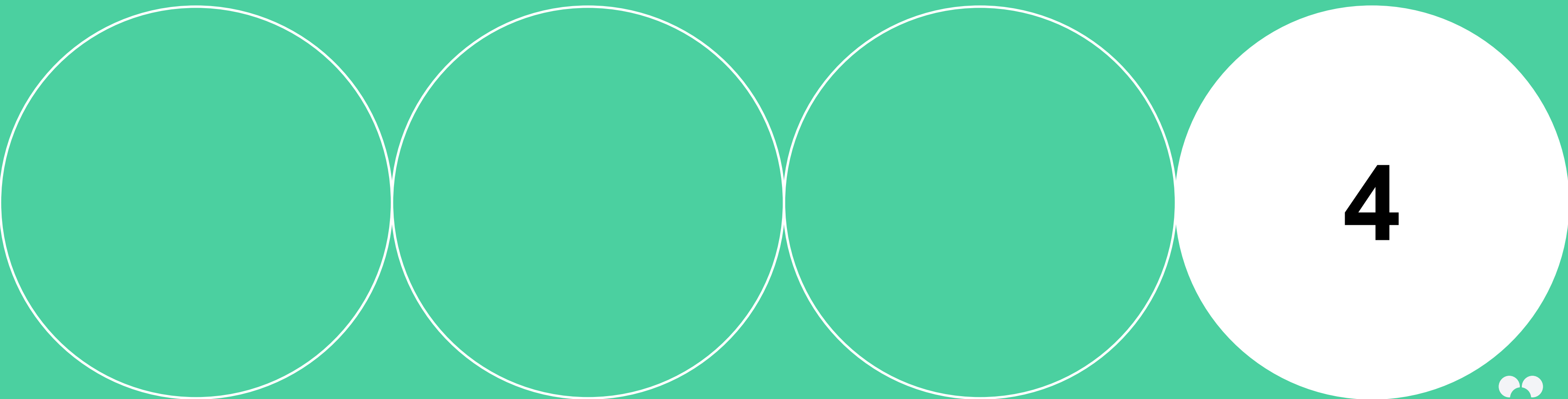


Kanban: возможности

- Исключение излишнего накопления задач для команд, обеспечивающих качество продукта
- Перераспределение ресурсов
- Повышение пропускной способности процесса разработки конвейерного потока с запросами на реализацию функций ПО



Scrum VS Kanban



Сходства

- Основаны на принципах Lean и Agile
- Ограничивают ИЗР
- Ориентированы на ранние и частые поставки продукта
- Полагаются на самоорганизующиеся команды
- Требуют декомпозиции задач
- Оптимизируют план релиза на основе эмпирических данных (производительности/времени выполнения задачи)



Отличия

Scrum

Как основная метрика для планирования и улучшения процессов используется производительность

НЗР ограничивается косвенно (за спринт)

Нельзя добавлять задачи в текущую итерацию

Предписаны 3 роли: Product Owner, Scrum Master, команда

Scrum-доска очищается между спринтами

Kanban

Как основная метрика для планирования и улучшения процессов используется время выполнения задачи

НЗР ограничивается явно (по статусам)

Можно добавлять новые задачи, когда это возможно

Нет предписанных ролей

Kanban-доска является неизменной



Итоги занятия

1

Методология Scrum

2

**Методология eXtreme
programming**

3

Принципы Kanban

4

Kanban vs Scrum



Домашнее задание



Цель

Закрепить знания о рассмотренных гибких методологиях разработки ПО

Задание

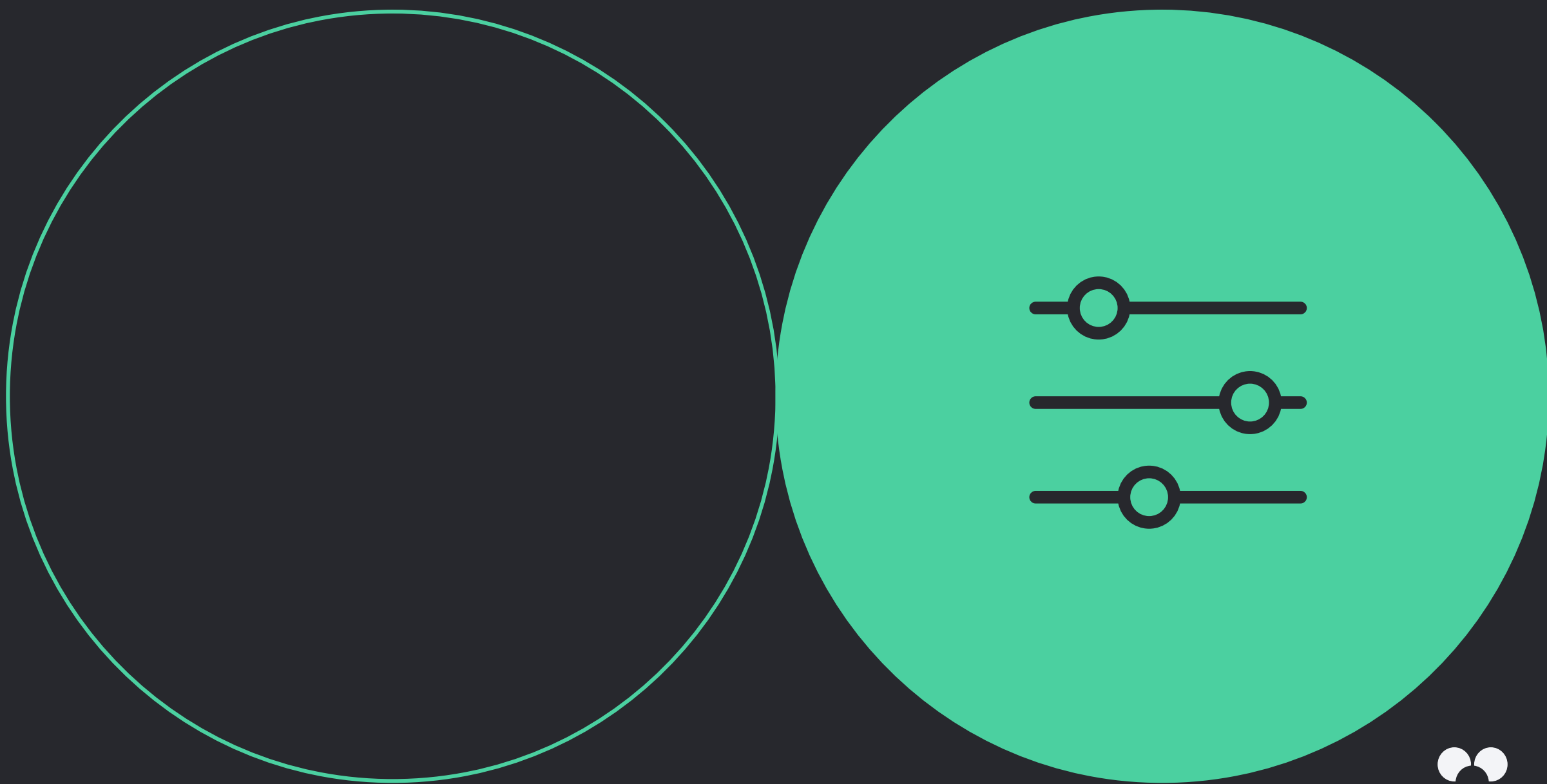
Пройти тестирование

Результат

Тест пройден



Дополнительные ресурсы



Что почитать, посмотреть, вдохновиться

1. [Хенрик Книберг "Scrum и XP: заметки с передовой"](#)
2. [Хенрик Книберг, Маттиас Скарин Scrum И Kanban: выжимаем максимум](#)
3. [Scrum Guide](#)
4. [Оценка задач в Story Points](#)
5. [Роман Короленко "STATIK - что это такое в контексте Kanban-метода"](#)
6. [kanban.club](#)



Гибкие методологии разработки ПО

Часть 2

Ксения Шипина
Системный аналитик

