

ЛЕКЦИЯ 5. ГИБКИЕ МЕТОДОЛОГИИ РАЗРАБОТКИ ПО

«Agile Model»

(гибкая методология разработки)

Scrum

XP (extremal programming)

– экстремальное программирование

RAD (Rapid Application Development)

– быстрая разработка приложений

Crystal

FDD (Feature Driven Development)

TDD (Test Driven Development)

Agile - принципы

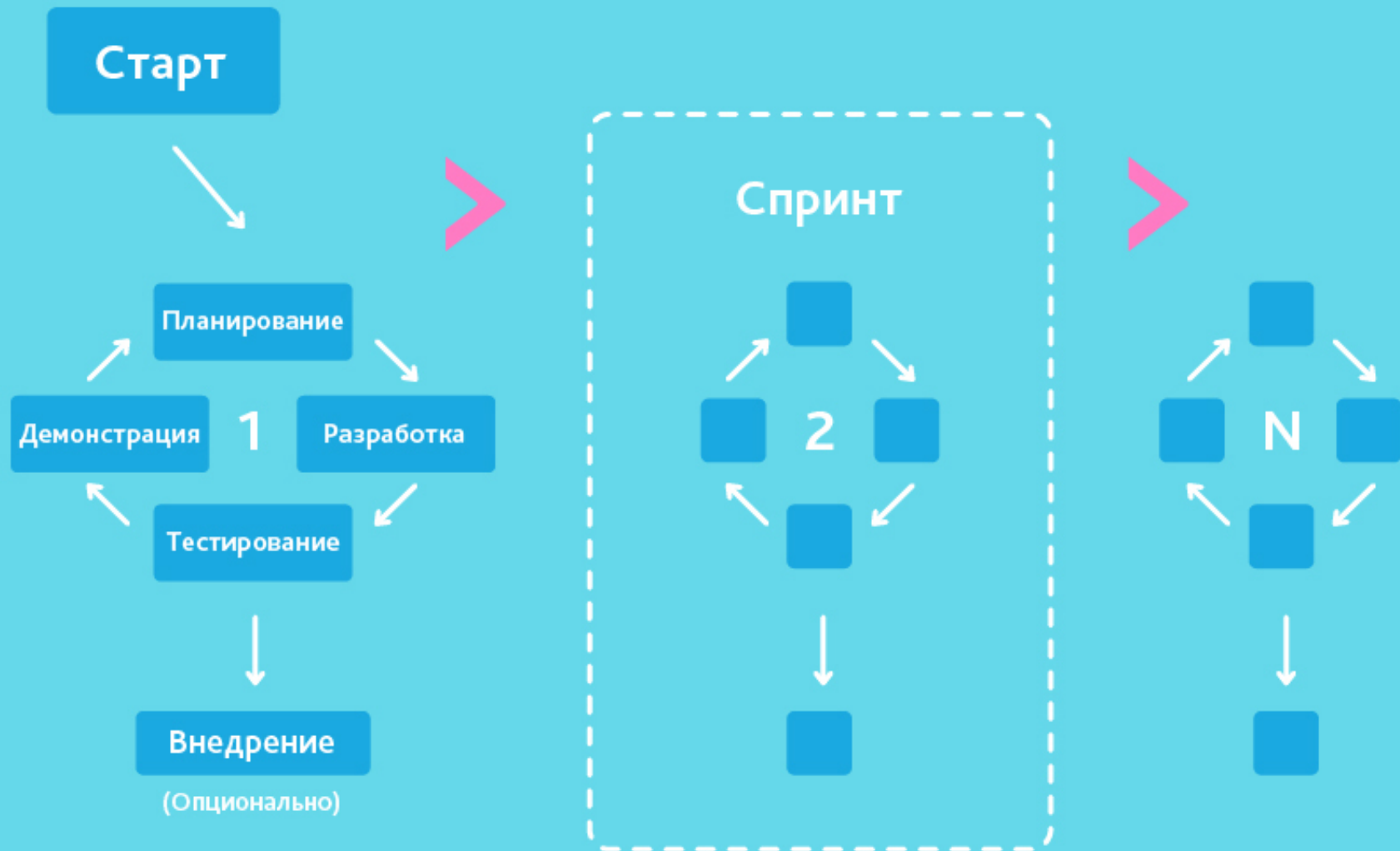
- ✓ Люди и процессы важнее инструментов
- ✓ Работающий продукт важнее документации
- ✓ Общение с заказчиком важнее контракта с ним (как документа)
- ✓ Изменчивость продукта важнее следования заранее установленному плану

Гибкая методология - когда использовать?

- Когда потребности пользователей постоянно меняются в динамическом бизнесе.
- Изменения на Agile реализуются за меньшую цену из-за частых инкрементов.
- В отличие от модели водопада, в гибкой модели для старта проекта достаточно лишь небольшого планирования.

«Agile Model»

(гибкая методология разработки)



«Agile Model»

(гибкая методология разработки)

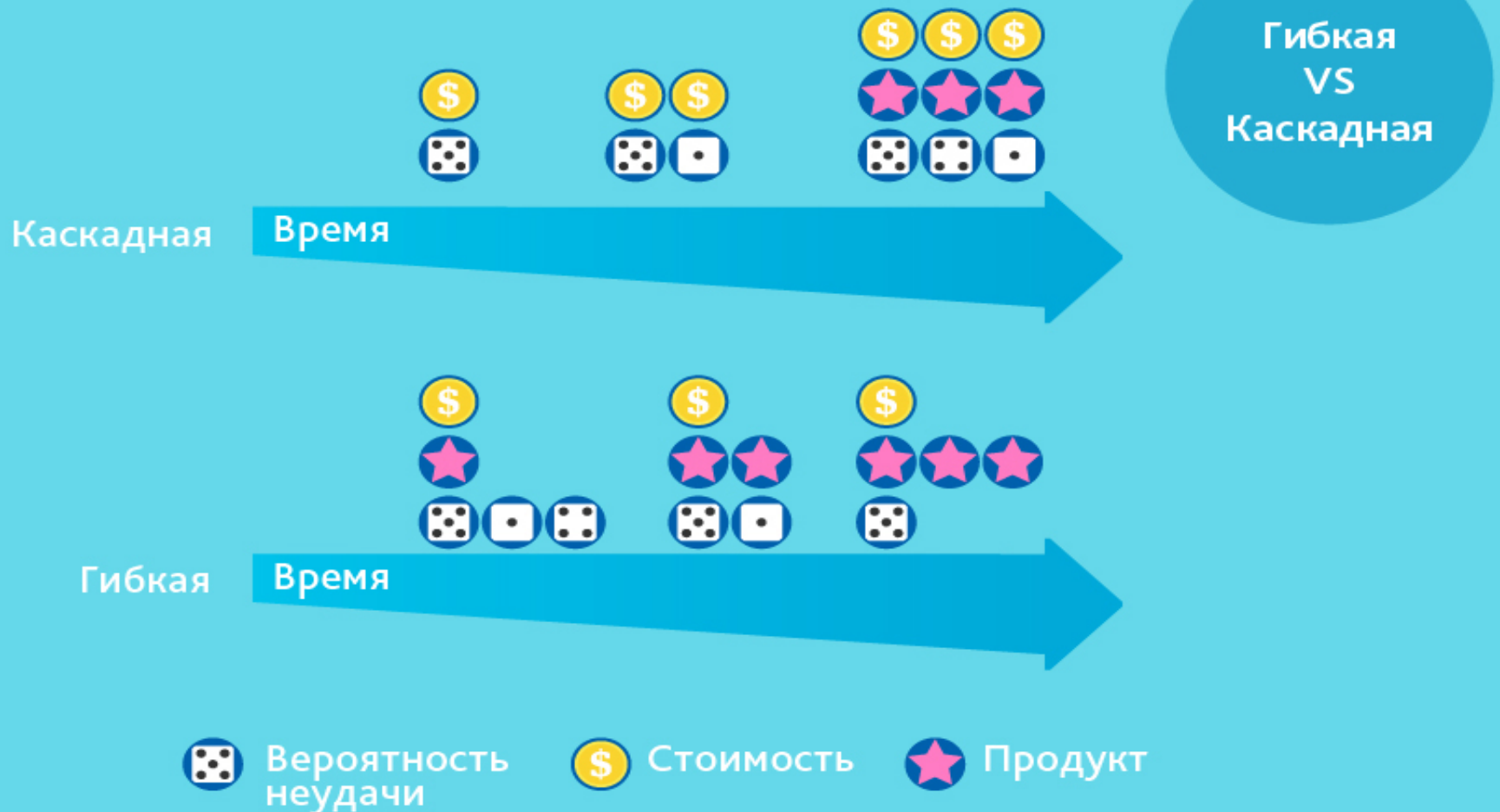
Преимущества:

- подходит для больших или нацеленных на длительный жизненный цикл проектов, постоянно адаптируемых к условиям рынка;
- лучше всего подходит для руководителей, которым свойственно генерировать, выдавать и опробовать новые идеи еженедельно или даже ежедневно;
- после каждой итерации заказчик может наблюдать результат и понимать, удовлетворяет он его или нет.

Недостатки:

- из-за отсутствия конкретных формулировок результатов сложно оценить трудозатраты и стоимость, требуемые на разработку.

Сравнение каскадной и гибкой моделей

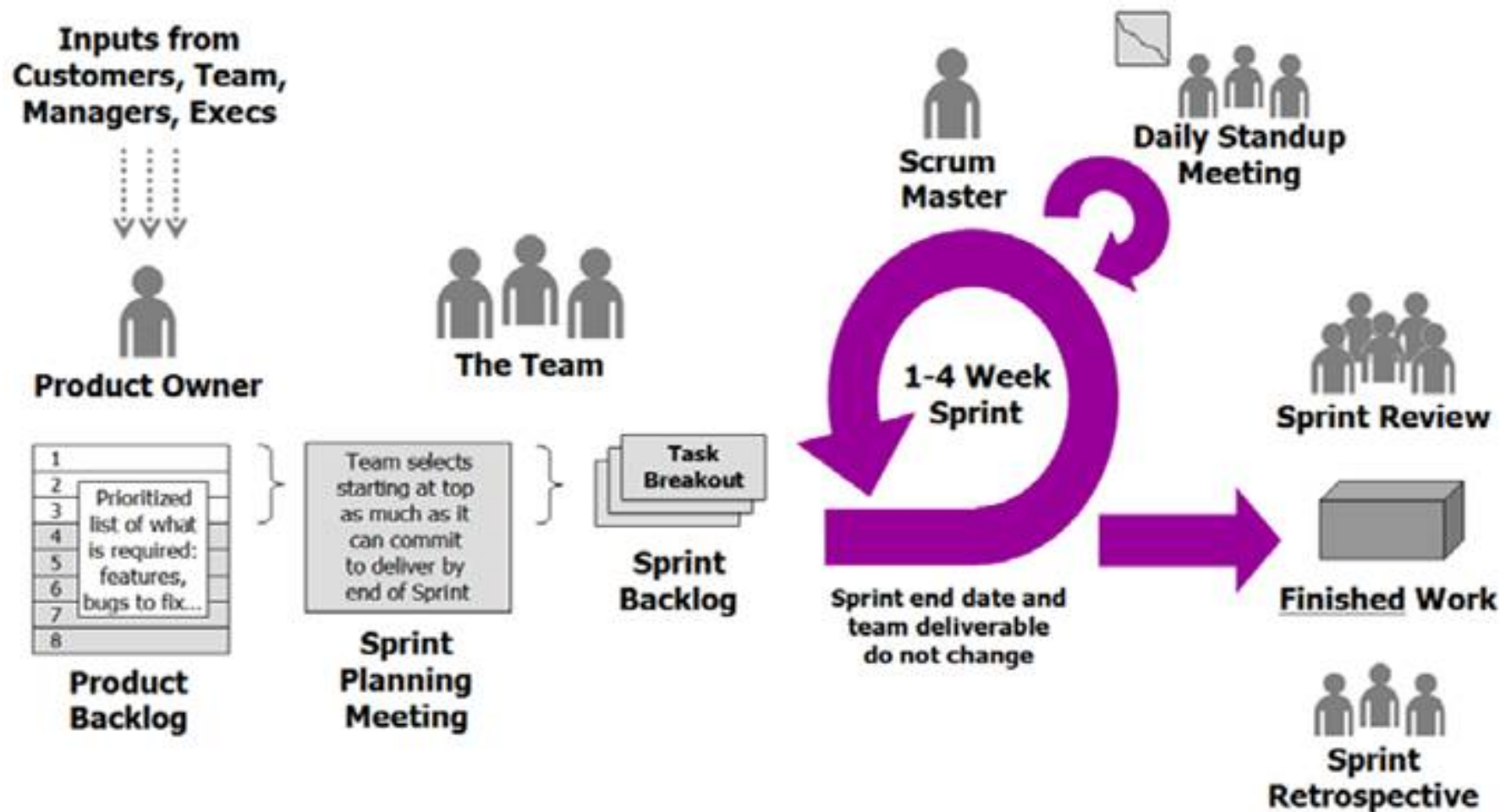


Scrum - кратко

- ✓ Scrum – популярный фреймворк для управления веб-проектами (*Hiroataka Takeuchi, Ikujiro Nonaka, 1986*)
- ✓ Является одной из гибких (Agile) методологий, в числе которых Kanban, XP
- ✓ Сам термин обозначает схватку в игре Регби



Scrum – крупным планом



Роль - Product owner

- ✓ «Владелец» продукта – менеджер проекта
- ✓ Управляет желаниями Клиента, пользователей, других менеджеров
- ✓ Защищает команду от лишних требований пользователей и стекхолдеров
- ✓ Управляет списком требований – Product Backlog

Роль – Scrum master

- ✓ «Судья», арбитр
- ✓ Устраняет препятствия перед командой
- ✓ Следит, чтобы все следовали правилам Scrum
- ✓ Управляет списком требований, приоритетами – Product backlog

Роль – Team member

- ✓ Такой роли нет на самом деле
- ✓ «Безликий» участник проектной команды
- ✓ Команда кроссфункциональна
- ✓ Предполагается тесное сотрудничество
- ✓ Конкретных ответственных – нет, есть «команда»

Product backlog

- ✓ Приоритезированный список требований

ID	User Story	Backlog Priority	Estimated Effort
1000	Search for resumes by keywords	1	s
1001	Enter resume online	2	m
1002	Post a job opening	3	l
1008	Add social network	4	l
1009	Post job opening to my social networks	5	m
1016	Add recruiter profile	6	m
1017	Add rating to a recruiter	7	s
1025	Review skill suggestions	8	s
1026	Approve a skill suggestion	9	s
1003	Find job openings that match my skills	10	m
1004	Find resumes with skills that match a job opening	11	s
1006	Search job openings by location	12	l
1010	Add alert for job postings on my social networks	13	l
1011	Filter job openings by industry type	14	m
1012	Upload resume in Word format	15	l
1015	Preview my resume with different templates	16	s
1018	Browse recruiter profiles by rating	17	s
1019	Add comments to a recruiter	18	m
1020	Browse recruiter profiles by location	19	m
1022	Associate a skill with my resume	20	l
1024	Suggest a new skill	21	l
1005	Filter job openings by skills	22	l
1007	Filter job openings by salary range	23	s
1013	Download resume in PDF format	24	l
1014	Enter resume online using a template	25	m
1021	Browse recruiter profiles by industry	26	m
1023	Associate a skill with a job opening	27	s

Sprint planning meeting

- ✓ Участники выбирают и оценивают задачи на Sprint. Присутствуют все.



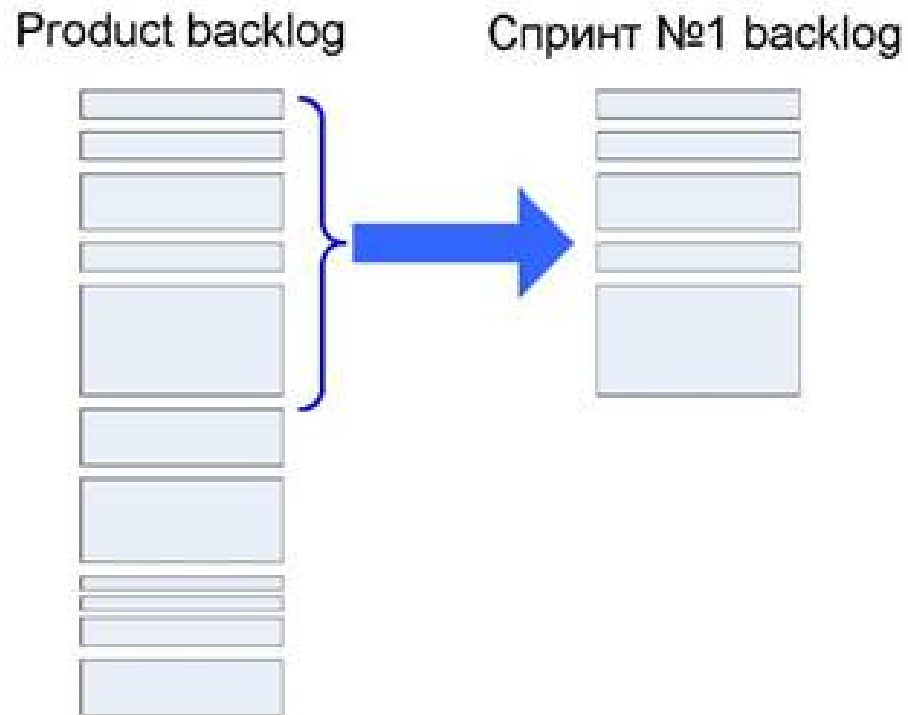
Sprint planning meeting

- ✓ Разработчики оценивают задачи, играя в Planning Poker



Sprint backlog

- ✓ Список задач на Sprint

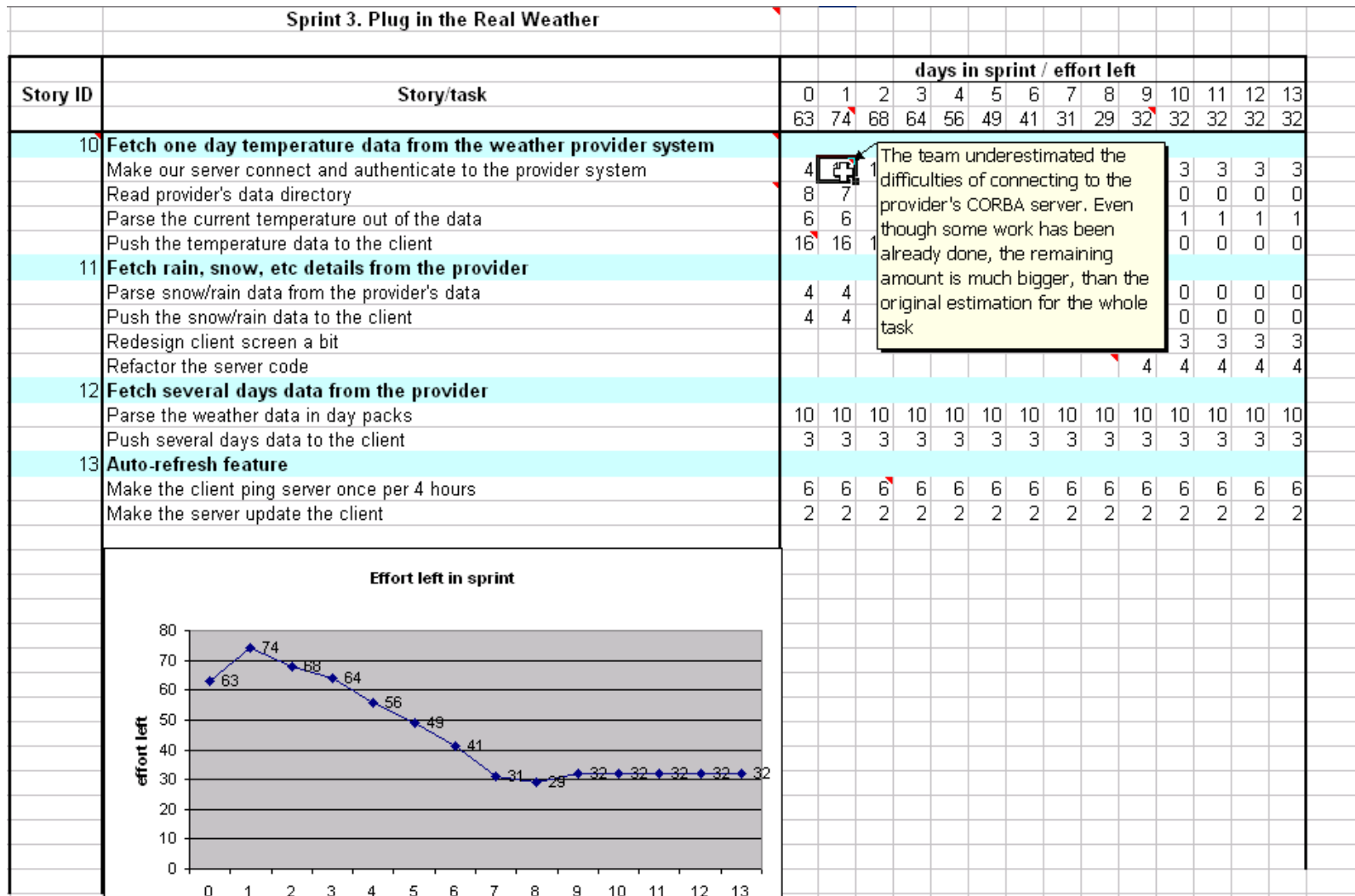


Sprint backlog

✓ Список задач на Sprint

Story	To Do		In Process	To Verify	Done
As a user, I... 8 points	Code the... 9	Test the... 8	Code the... DC 4	Test the... SC 6	Code the... D
	Code the... 2	Code the... 8	Test the... SC 8		Test the... SC 8
	Test the... 8	Test the... 4			Test the... SC 6
As a user, I... 5 points	Code the... 8	Test the... 8	Code the... DC 8		Test the... SC
	Code the... 4	Code the... 6			Test the... SC 6

Sprint backlog

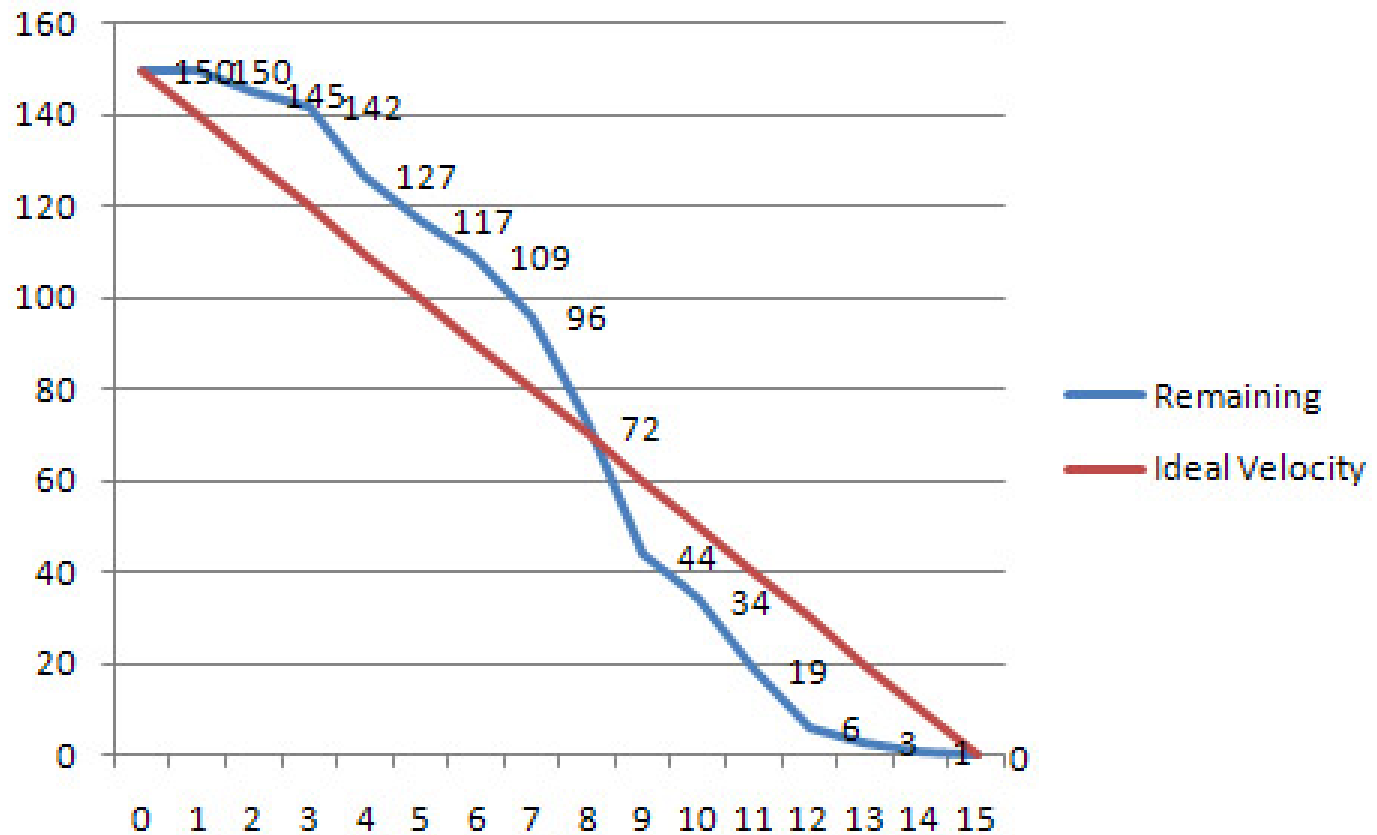


Daily standup meeting

- ✓ Утром или вечером
- ✓ Не более 15 минут
- ✓ Что было сделано вчера?
- ✓ Что будет сделано сегодня
- ✓ С какими проблемами кто столкнулся?



Burndown chart



Definition of Done (DoD)

Условие готовности задачи – очень тонкий момент

- ✓ Как задача попадает в «готовые»?
- ✓ Участие тестировщиков
- ✓ Ответственность разработчиков
- ✓ Модульное тестирование

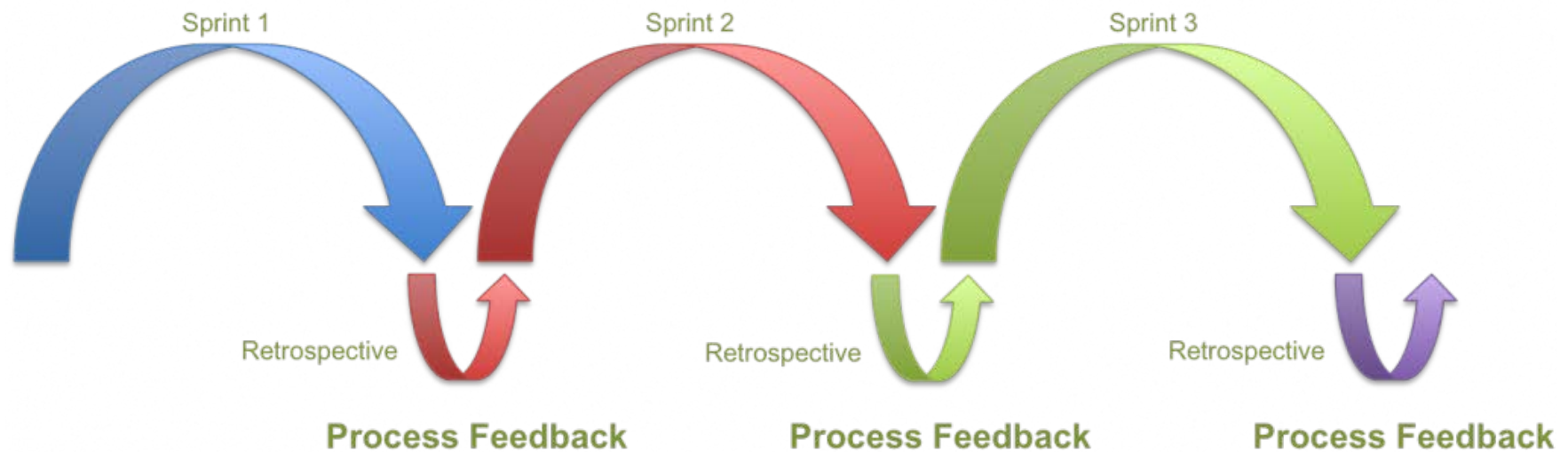
Sprint review meeting

- ✓ Демонстрация результатов спринта Клиенту



Sprint retrospective

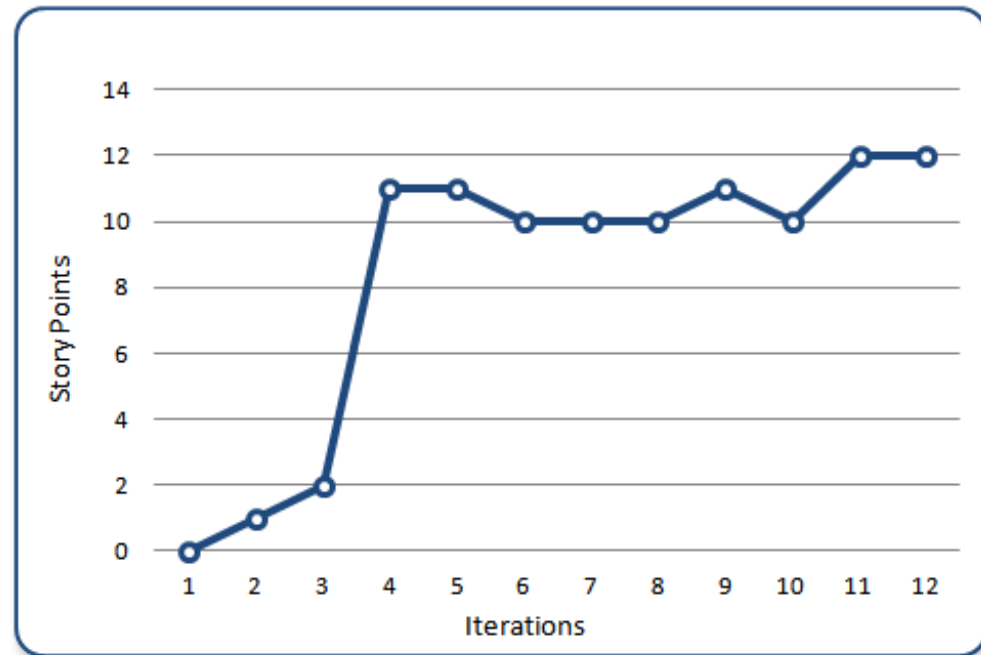
- ✓ Что вызывало проблемы в спринте
- ✓ Что можно сделать лучше?



Velocity

Скорость работы команды

- ✓ Измеряется для каждого завершенного спринта
- ✓ Прогнозируем сроки релизов
- ✓ Оптимизируем процесс



Нужен ли вам Scrum?

Плюсы:

- ✓ Просто внедрить
- ✓ Разработчикам обычно нравится
- ✓ Прозрачность проекта
- ✓ Ориентация на результат

Минусы:

- ✓ Не все люди сработаются
- ✓ Издержки на «болтовню» 10-30%
- ✓ Качество веб-системы может сильно пострадать

Не заиграйтесь!

- ✓ Не теряйте контроль за Scrum-командой
- ✓ Строго следите за качеством выполнения



User story

ID — уникальный идентификатор, порядковый номер, применяемый для идентификации историй в случае их переименования.

Название (Name) — краткое описание истории. *Важность (Importance)* — степень важности данной истории, по мнению владельца продукта, полученная в результате переговоров с заказчиком. Чем больше значение, тем выше приоритет.

Бизнес-процесс (Business process) — краткое описание бизнес-процесса, в котором программисты бизнес-приложений получают в два раза больше программистов других приложений.

Начальная оценка (initial estimate) — оценка объёма работ, сделанная до начала спринта, необходимого для реализации истории. Измеряется в story point'ах. Приблизительно соответствует числу «идеальных человеко-часов».

Как продемонстрировать (how to demo) — краткое описание способа демонстрации завершённой задачей заказчику, скрам-мастеру и владельцу продукта в конце спринта. Данное поле может содержать собой название (идентификатор) или код автоматизированного теста для приёмо-сдаточного испытания.

Критерии приёмки (acceptance criteria) — значимые детали реализации истории, уточняющие требования владельца продукта, собранные всеми участниками SCRUM-команды при планировании спринта

Kanban

- Немного из истории
- Примеры Kanban для разработки ПО
- Work In Progress
 - Пример
 - Снижение WIP (закон Литтла)
 - Критерии готовности
- **Внеплановые задачи**

История Kanban

Kanban - система организации производства и снабжения, позволяющая реализовать принцип «точно в срок»:

- ▣ разработана фирмой «Тойота» в 1959 году
- ▣ в переводе с японского означает «карточка»

Принципы Kanban:

- ▣ визуализация (карточки)
- ▣ ограничение количества
- ▣ отсутствие брака
- ▣ формирование общей схемы

Основная задача:

- ▣ уменьшать количество «выполняющейся в данный момент работы»

Kanban в разработке ПО

- Не процесс, а система ценностей, как и SCRUM
- Уменьшение «выполняющейся в данный момент работы»
- Более «гибкая» методология, чем SCRUM:
 - Нет таймбоксов (ни на задачи, ни на спринты)
 - Задачи больше и их меньше
 - Оценки сроков на задачи опциональны или отсутствуют
 - «Скорость работы команды» отсутствует, считается среднее время на полную реализацию задачи

Kanban-доска

Backlog	Очередь 2	Разработка 3		Тестирование 2	Готово
		В процессе	Готово		
<div><div></div><div></div><div></div></div>					
<div><div></div><div></div><div></div></div>					
<div><div></div><div></div><div></div></div>					

Kanban-доска

Backlog	Очередь 2	Разработка 3		Тестирование 2	Готово
		В процессе	Готово		
	<div>1</div> <div>2</div>				

Kanban-доска



Kanban-доска

Backlog	Очередь 2	Разработка 3		Тестирование 2	Готово
		В процессе	Готово		
					
					

Kanban-доска



Канбан-доска

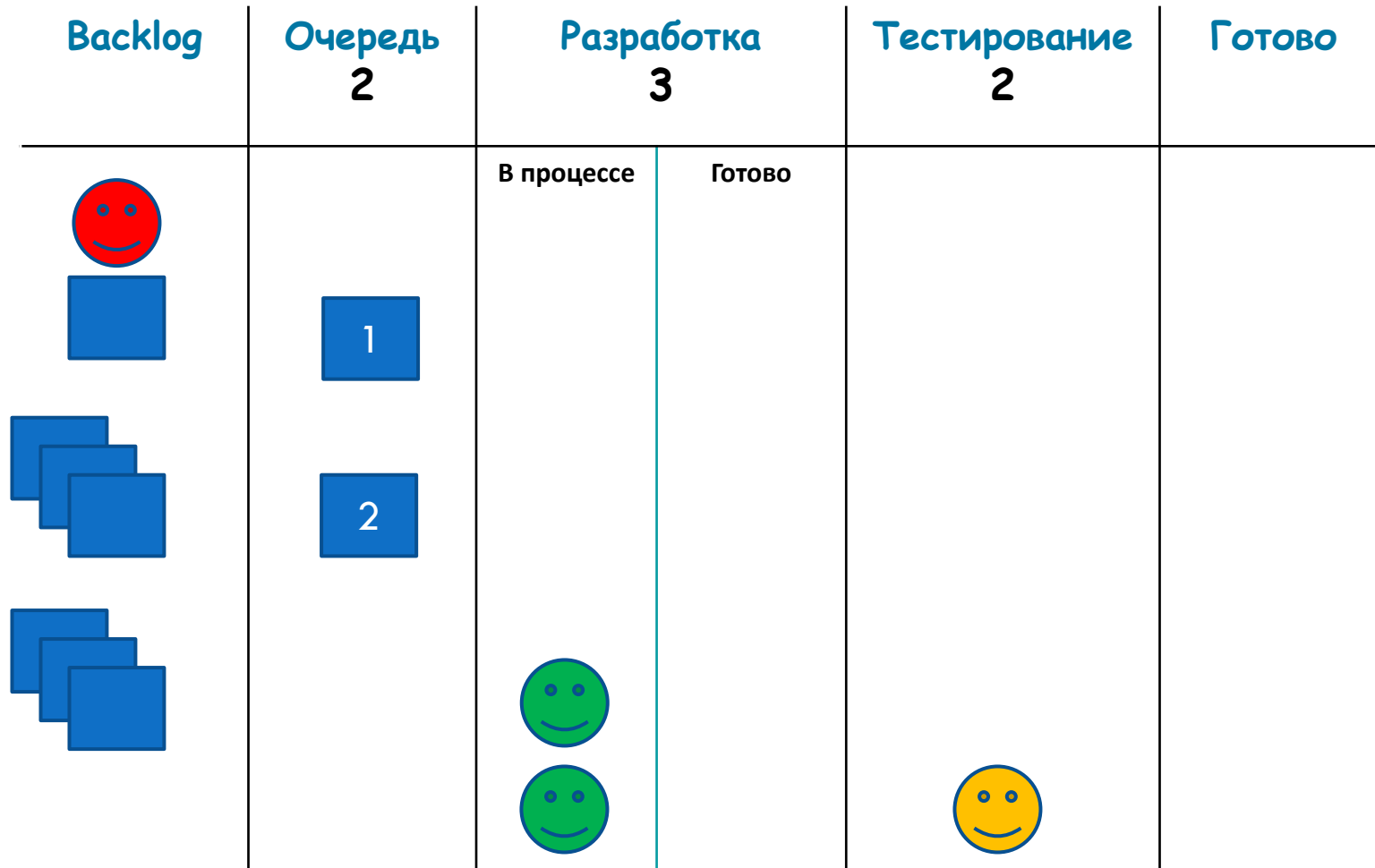
Backlog	Очередь 2	Разработка 3		Тестирование 2	Готово
		В процессе	Готово		
	5		3		1
	6	4		2	



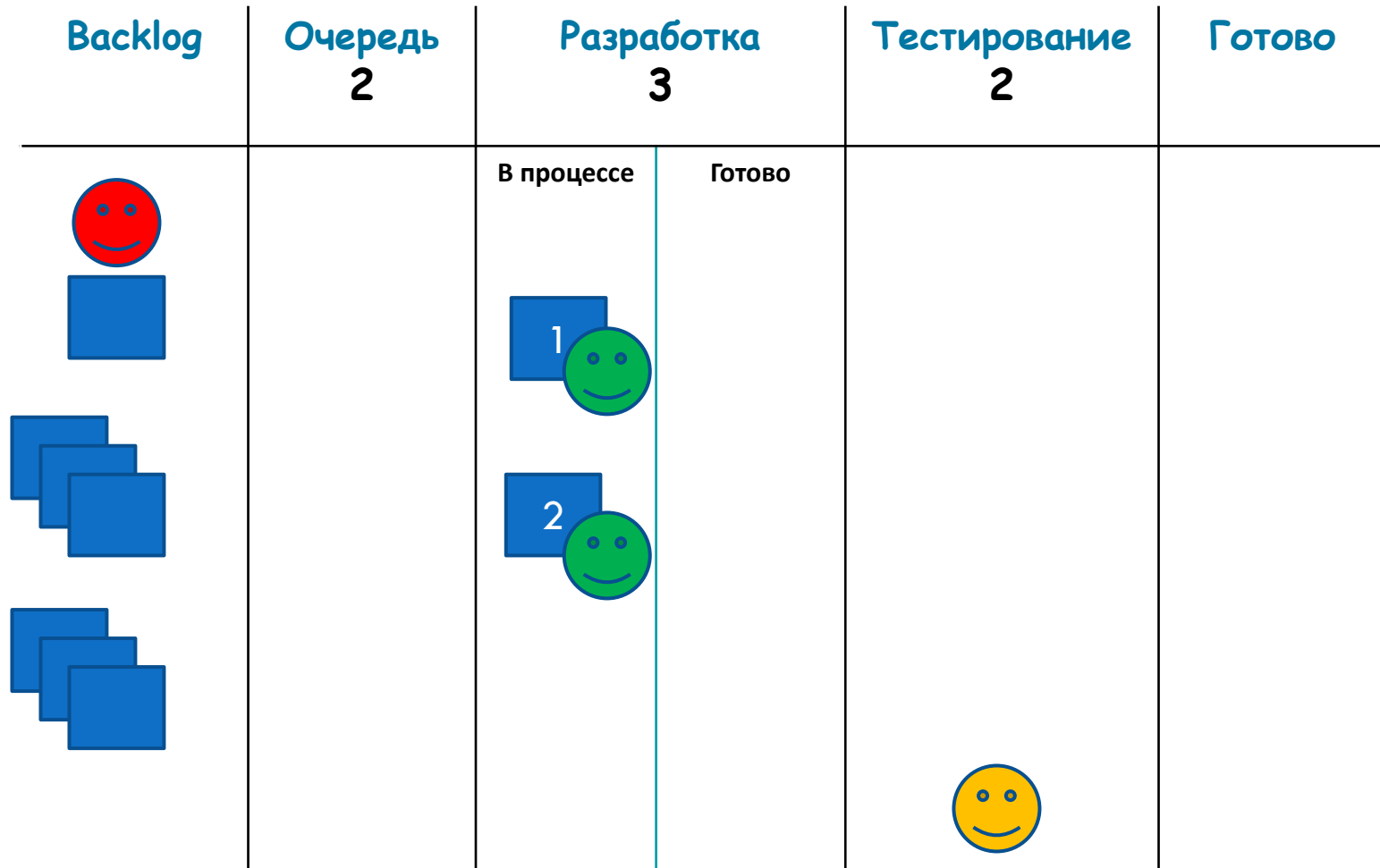
Work In Progress

Backlog	Очередь 2	Разработка 3		Тестирование 2	Готово
		В процессе	Готово		
					
					
					
		 			

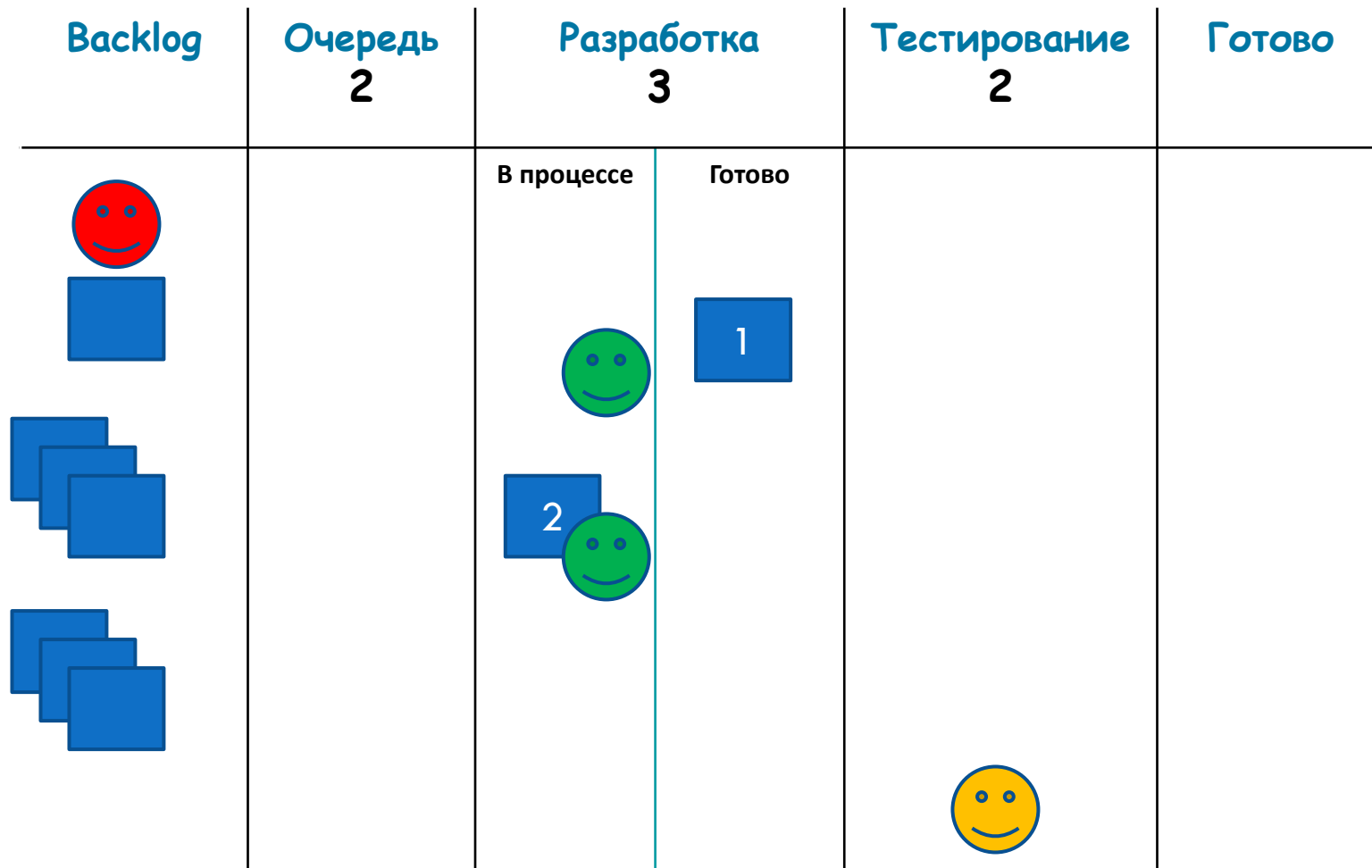
Work In Progress



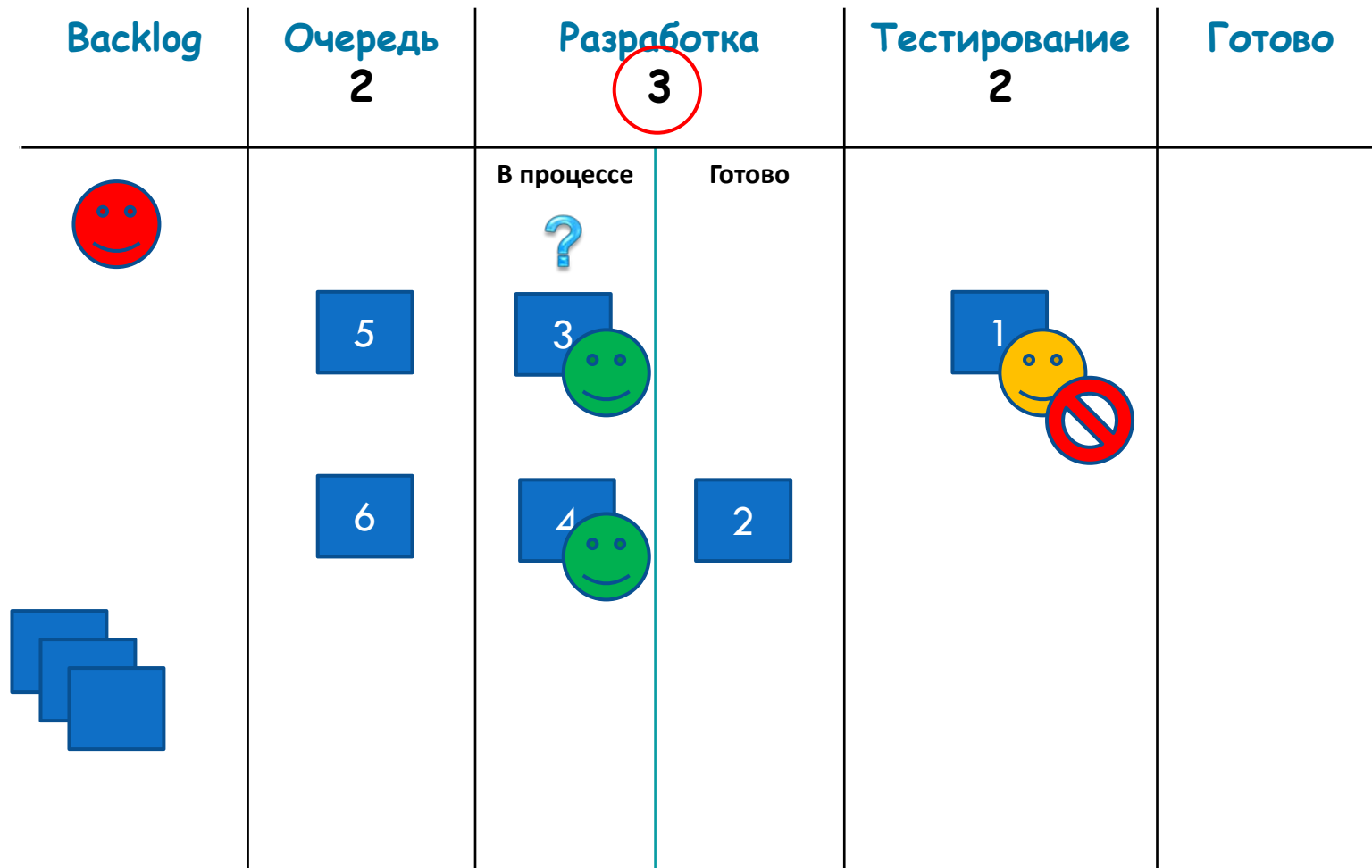
Work In Progress



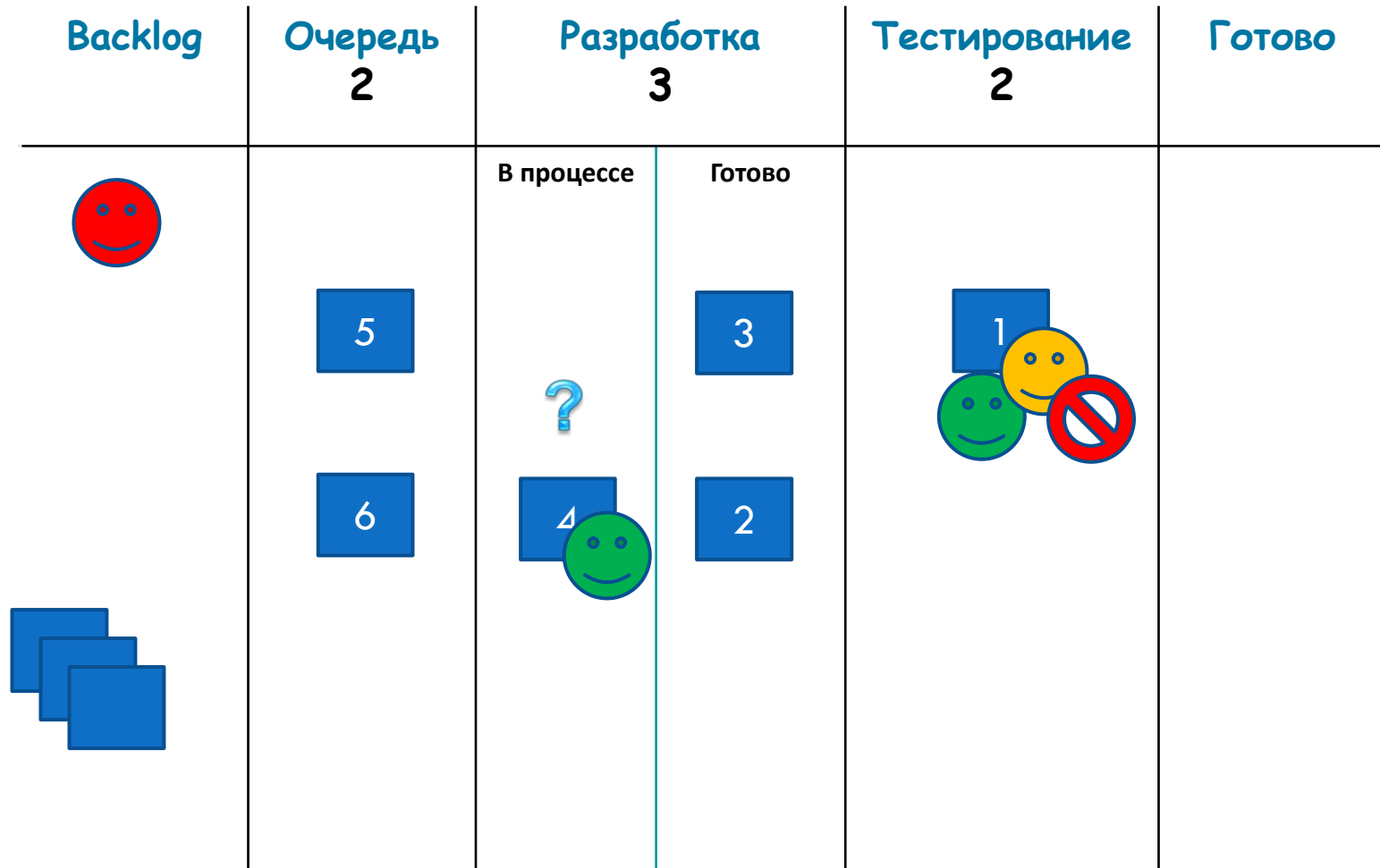
Work In Progress



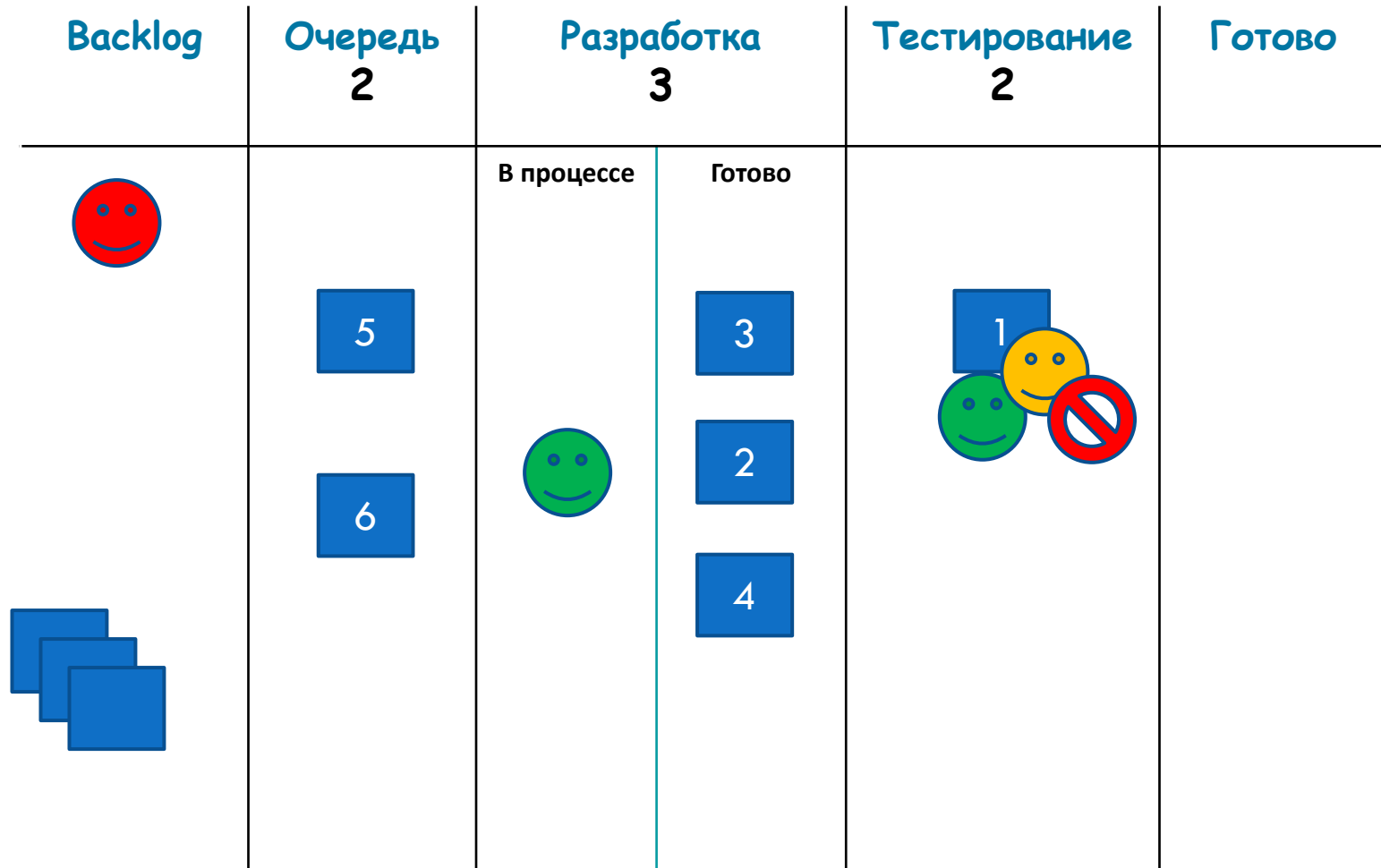
Work In Progress



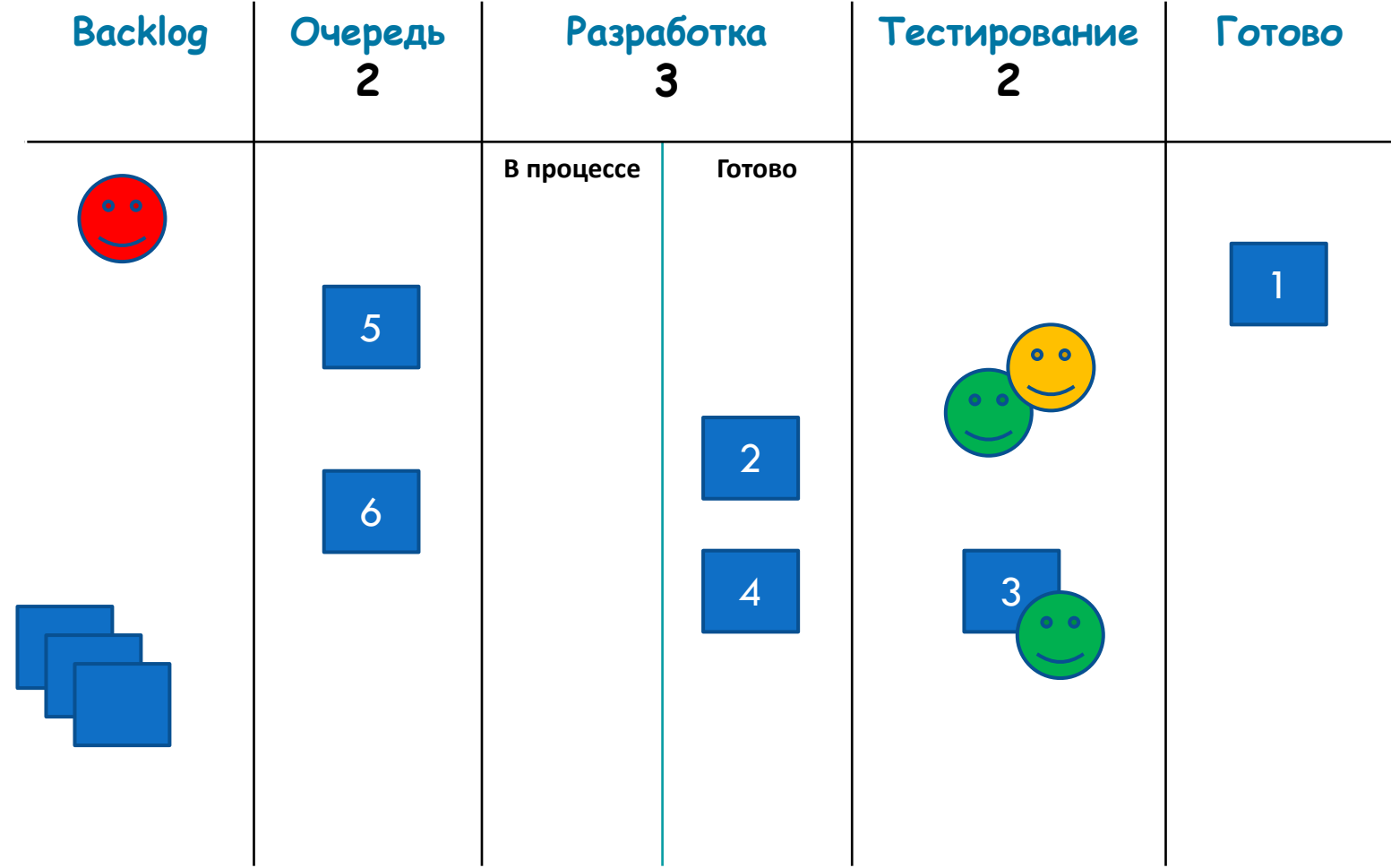
Work In Progress



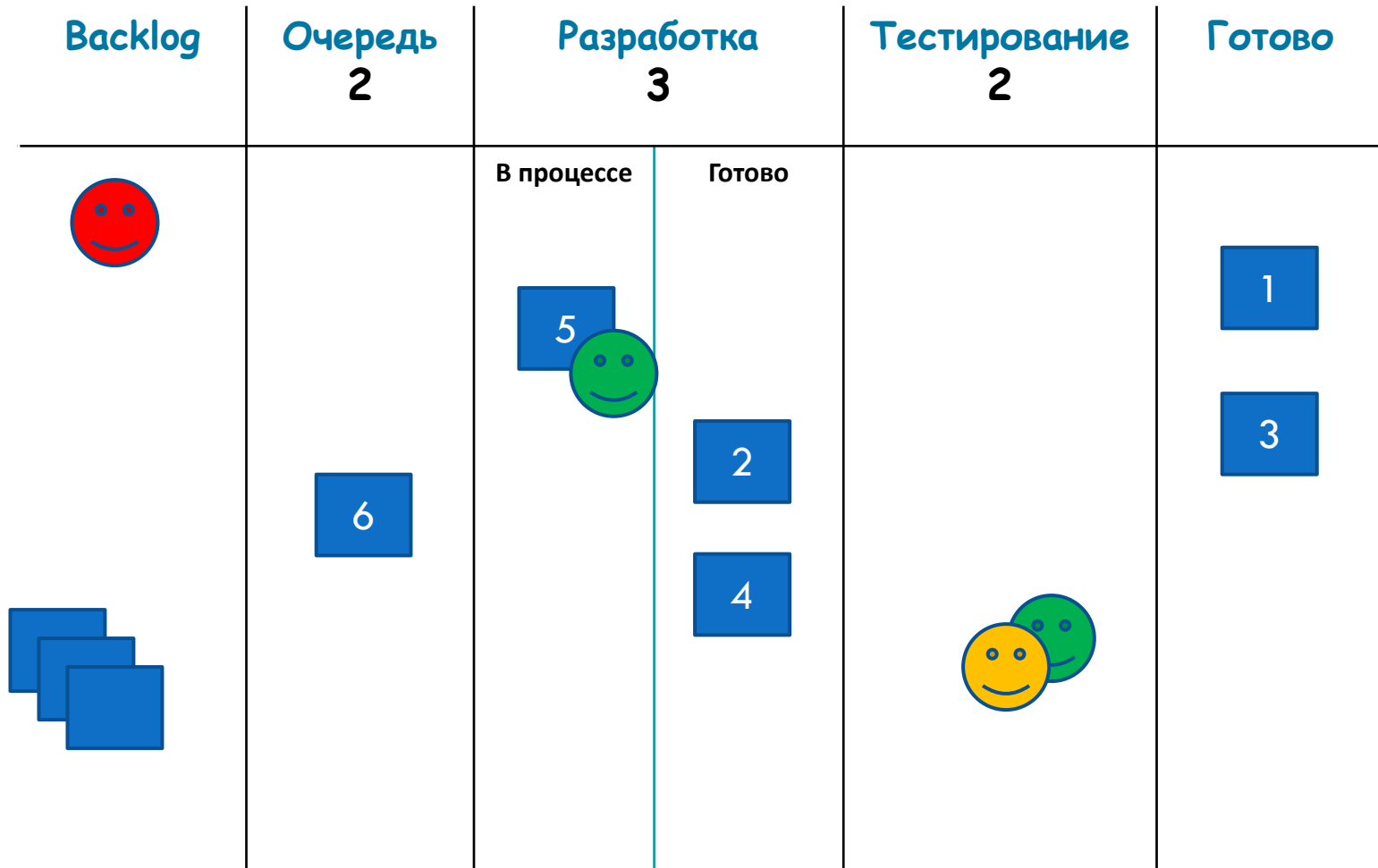
Work In Progress



Work In Progress



Work In Progress



Work In Progress

- Закон Литтла:

- Среднее время ожидания = размер очереди / скорость обслуживания
- Lead Time = WIP / Average Completion Rate
- 100 человек / 5 чел в час = 20 часов

Work In Progress

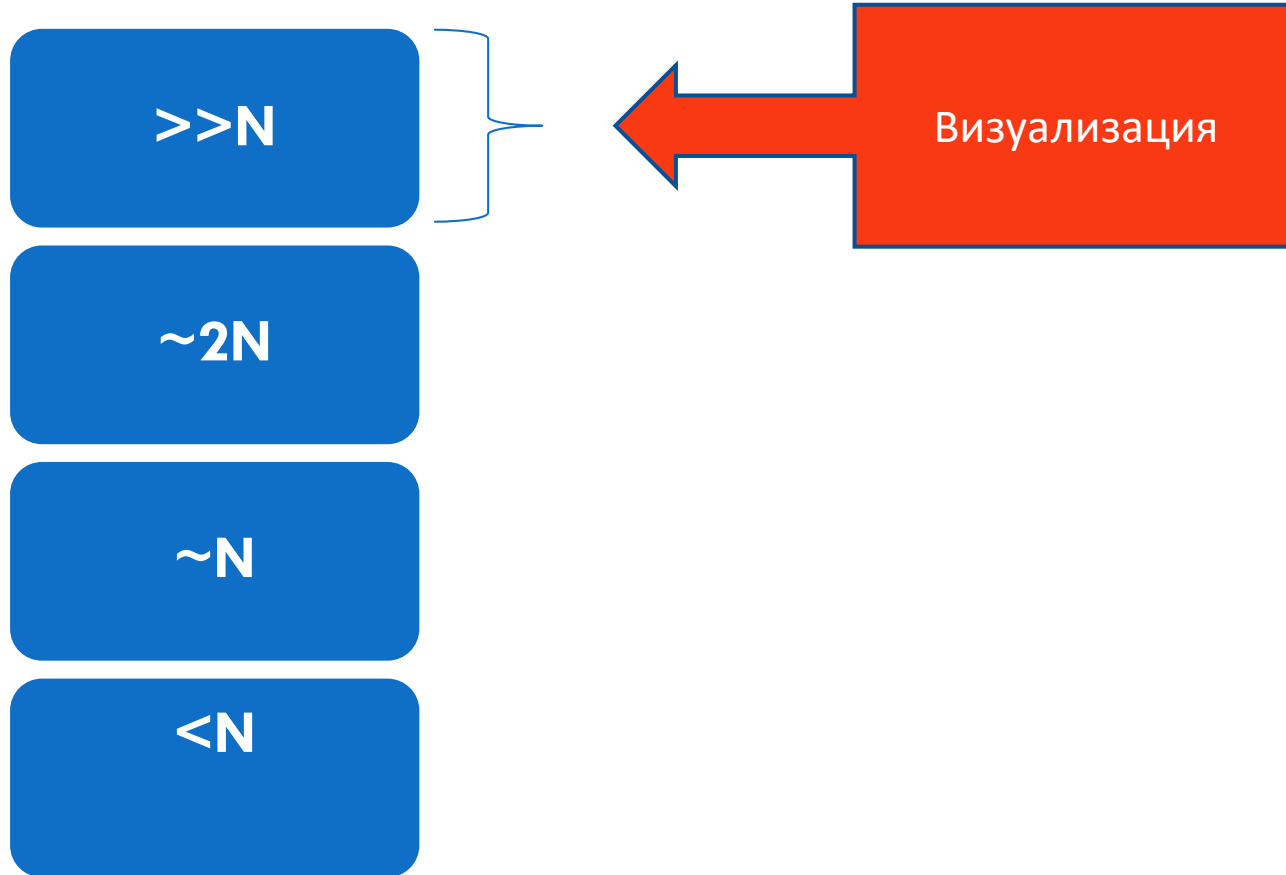
$>>N$

$\sim 2N$

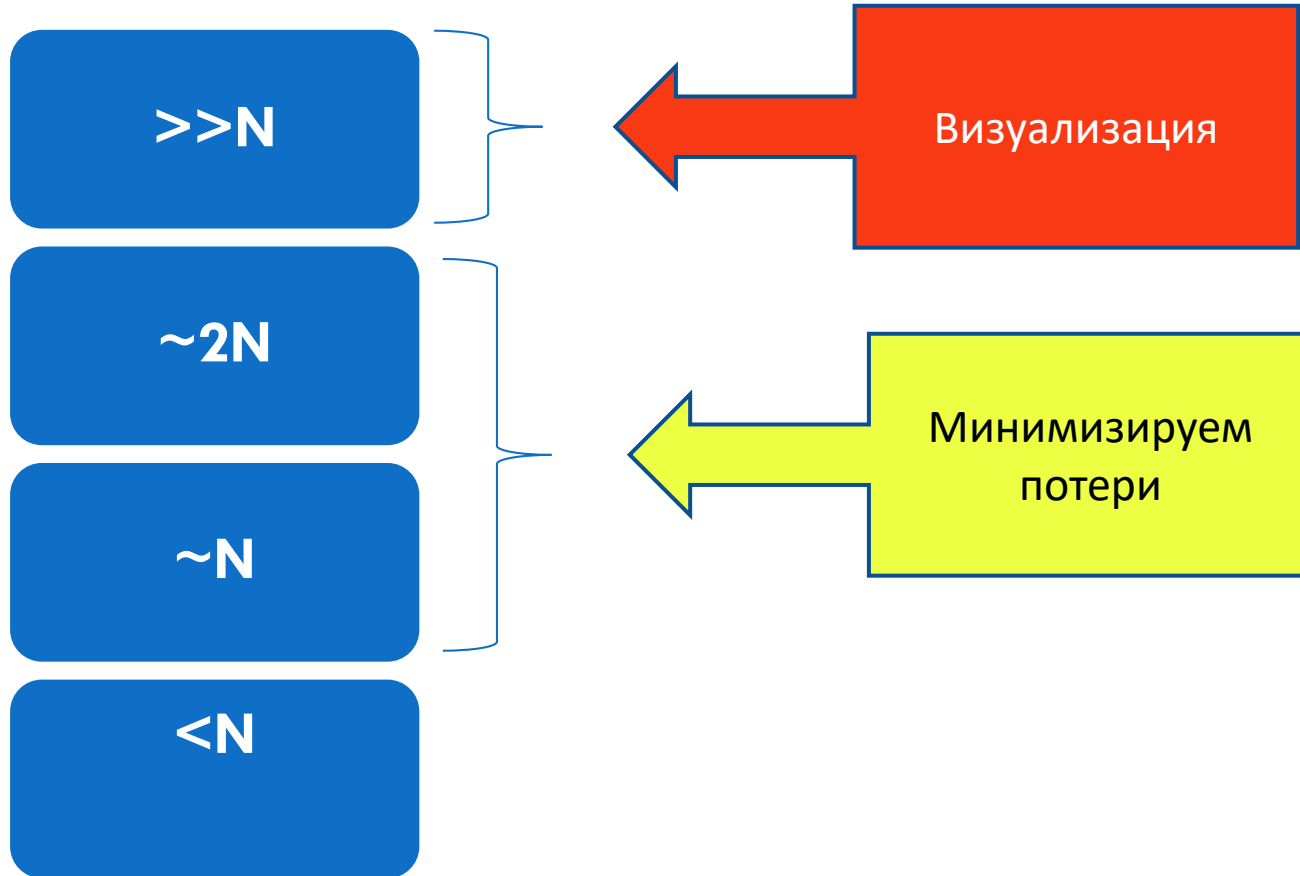
$\sim N$

$<N$

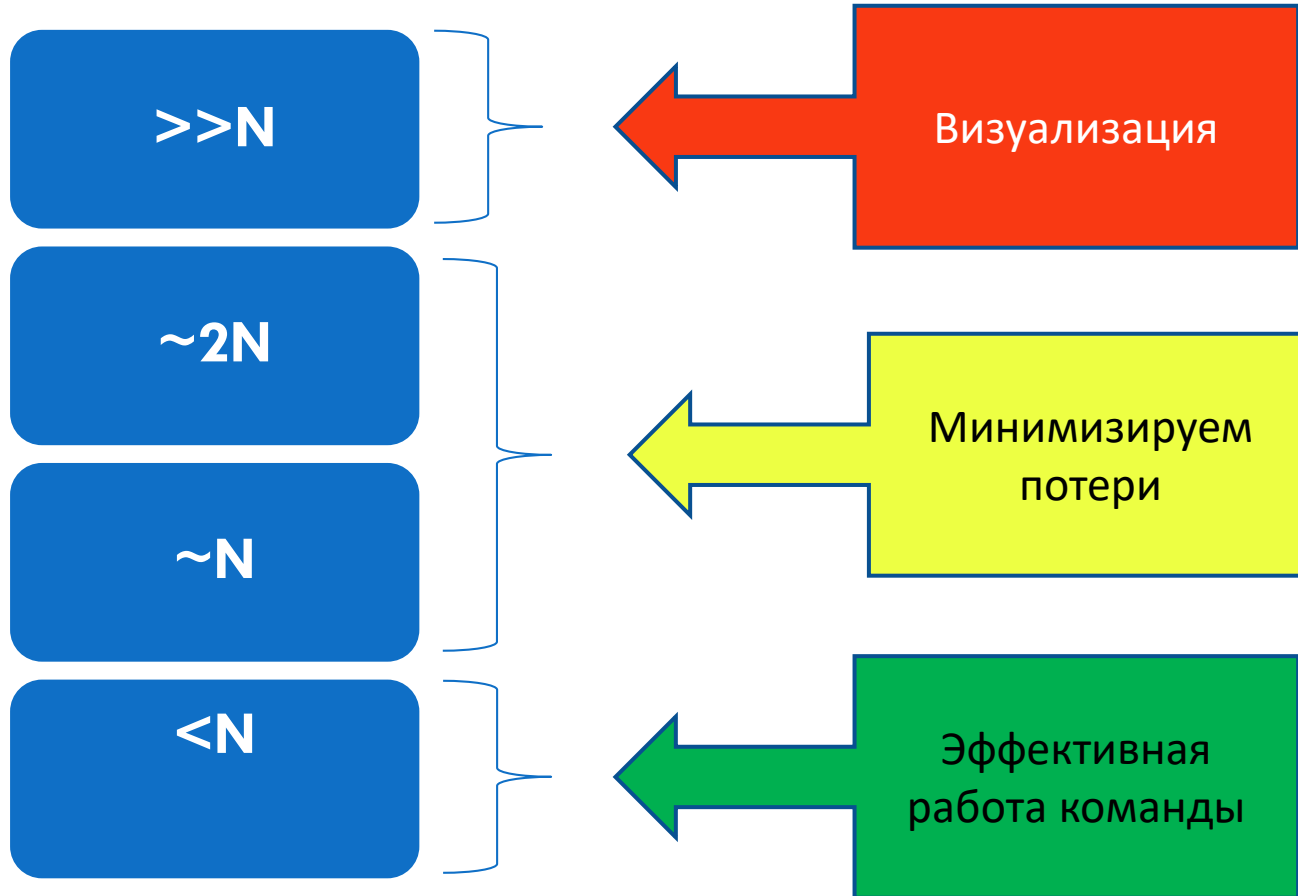
Work In Progress



Work In Progress



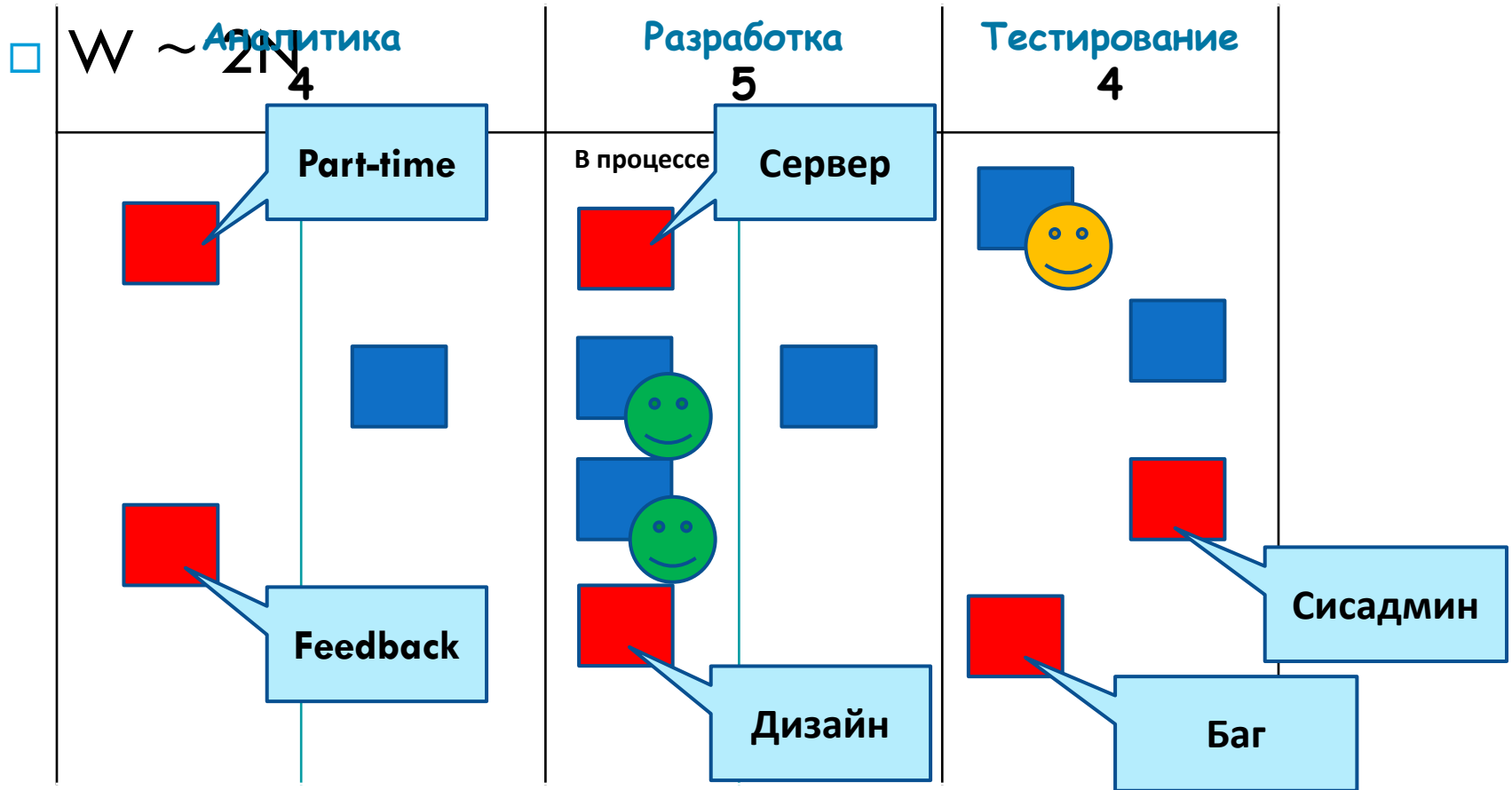
Work In Progress



Work In Progress

- $WIP \gg N$
 - Слабое взаимодействие в команде
 - Сопротивление применению Kanban
 - Желание заниматься только своими задачами

Work In Progress



Work In Progress

- $WIP \sim N$
 - Минимизация внешних зависимостей
 - Самоуправляемость команды (проблемы решаются внутри и быстро)
 - Все участники оперативно помогают друг другу

Work In Progress

- $WIP < N$
 - Командная работа
 - Нет узких мест
 - Кроссфункциональность звена разработки

Work In Progress

- Аналитика
 - ▣ Сценарий «демо» и приёмочные тесты
 - ▣ Заказчик (партнёр)
 - ▣ Ревью программистом
- ▣ **Разработка**
 - ▣ Тесты на основные методы
 - ▣ Код-ревью
 - ▣ Документация
- ▣ **Приемка**
 - ▣ Принял заказчик
 - ▣ Готово к деплою

Внеплановые задачи



Внеплановые задачи



- Приоритет

- ▣ «Срочный» баг



- ▣ Приоритетная задача



- ▣ Риск нарушения сроков



- ▣ Остальные в порядке поступления