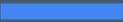
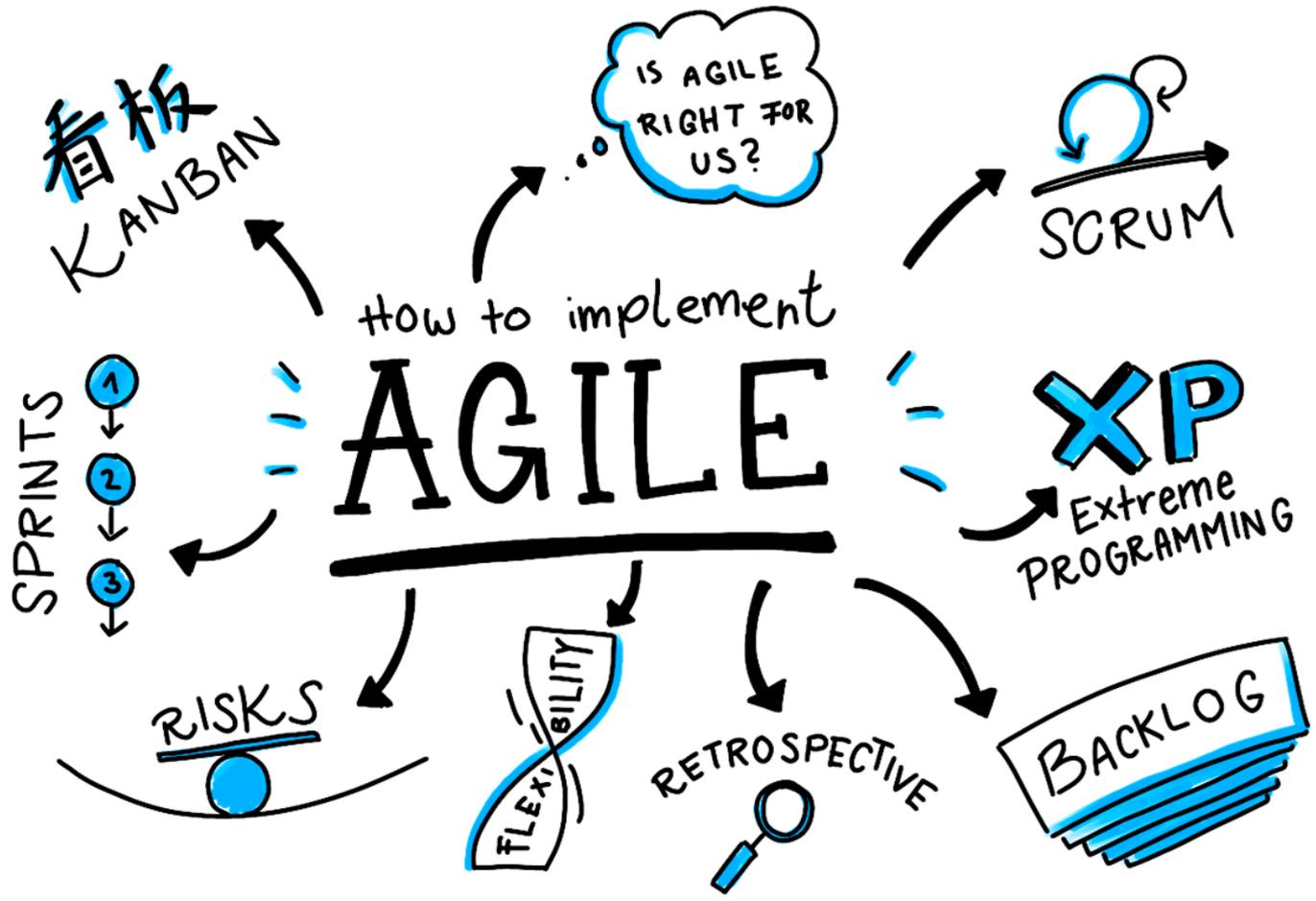


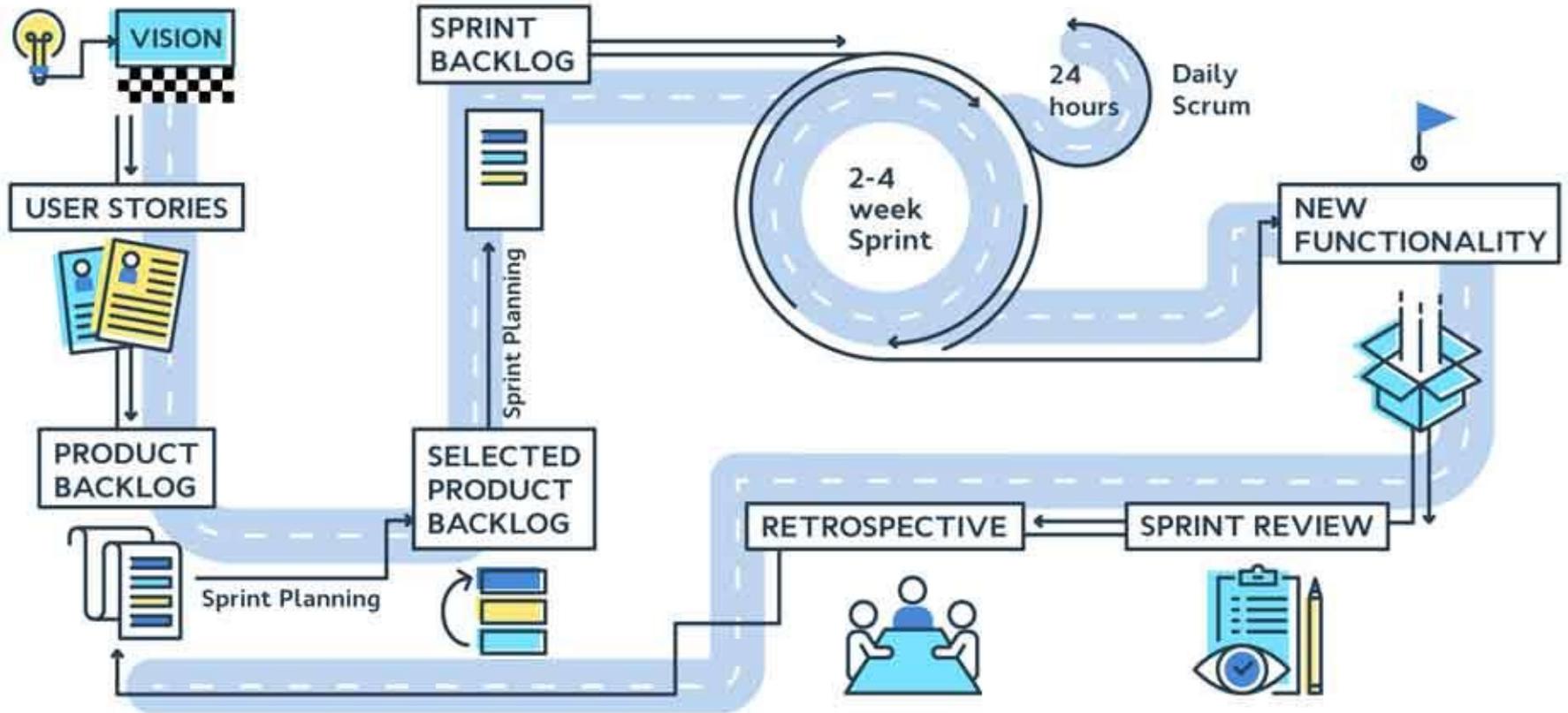
Роль аналитика в SCRUM



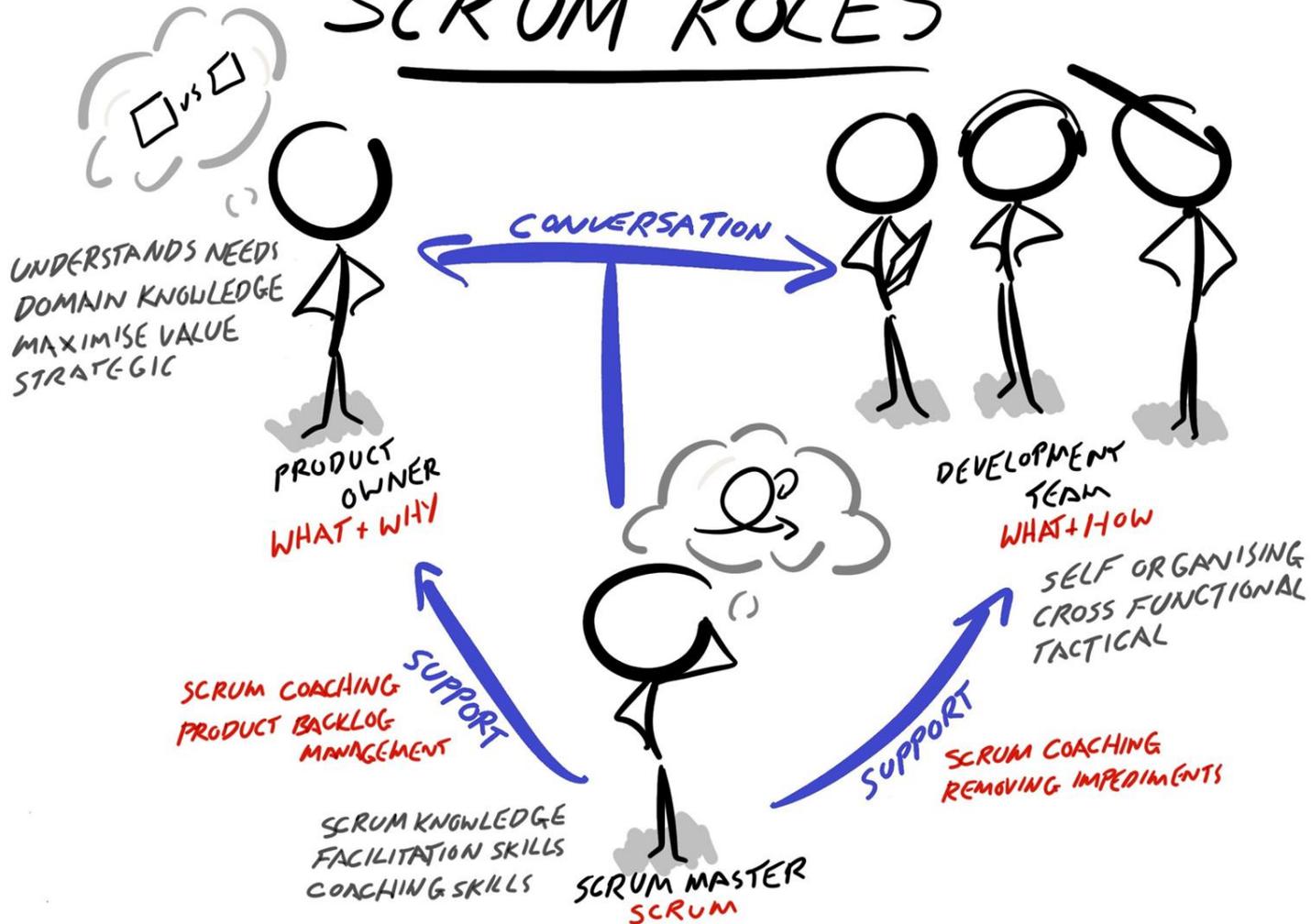
Уставщикова Юлия



SCRUM PROCESS



SCRUM ROLES



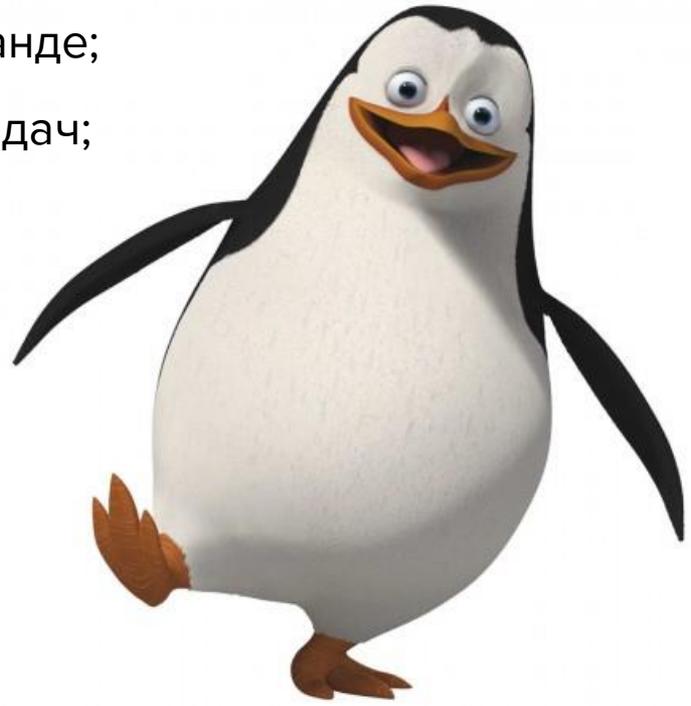
Product Owner

- Определение элементов Product Backlog;
- Правильное расположение элементов для оптимизации достижения цели;
- Обеспечение понятности и прозрачности Product Backlog;
- Обеспечение прозрачности и понятности требований, над которыми предстоит работать всей Scrum Team;
- Общая оптимизация для достижения наибольшей ценности работы Development Team;
- Ответственность за понимание бэклога командой разработки.



Scrum Master

- Устранение проблем, образующихся внутри команды;
- Выявление скрытых вопросов;
- Создание дружественных отношений в команде;
- Слежение за процессами и выполнением задач;
- Смена статусов задач в спринте;
- Проведение Daily Scrum Meeting;
- Организация встреч перед спринтами;
- Помощь Product Owner с Backlog.

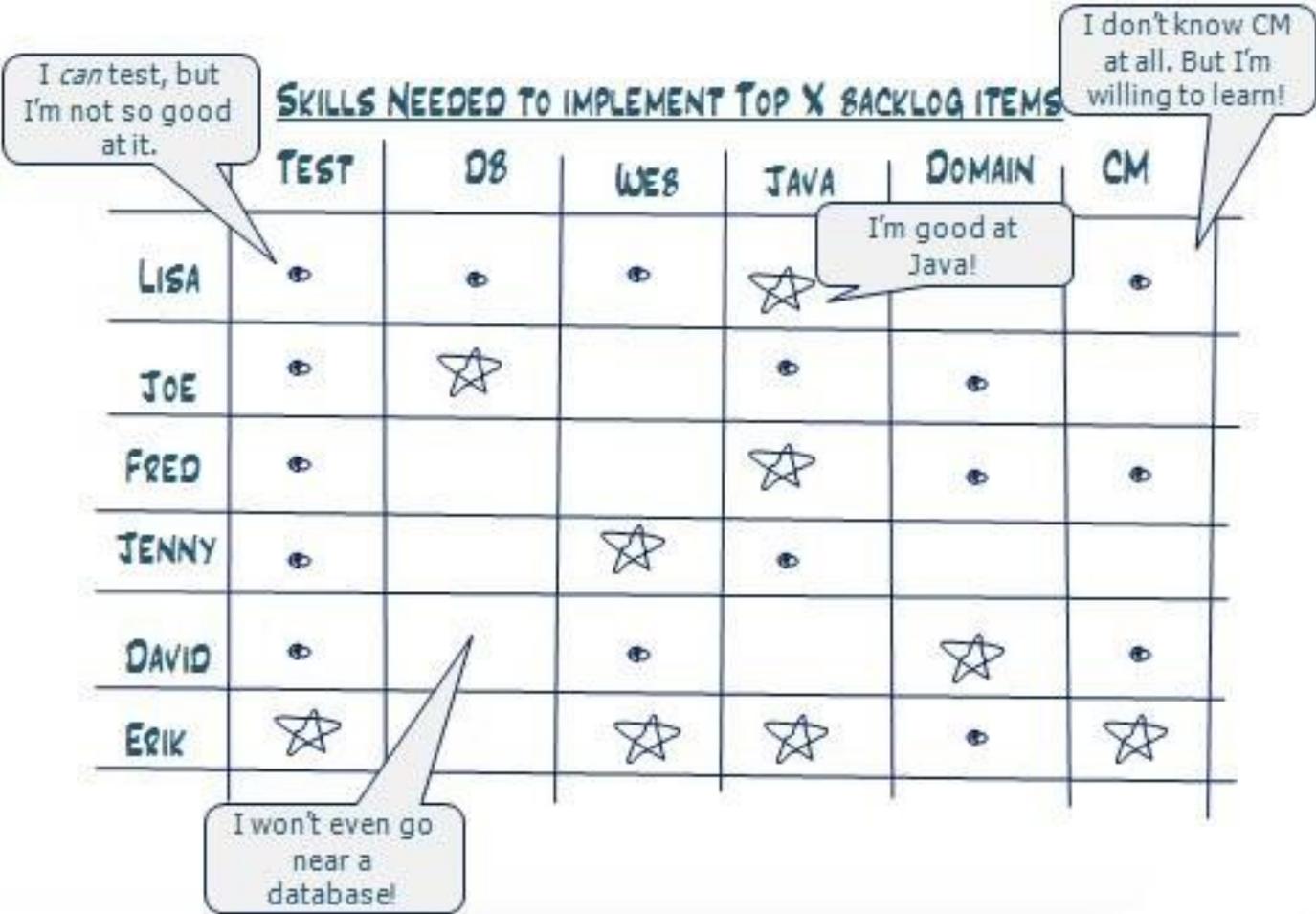


Команда разработки

- Команда полностью самоорганизована. На её работу никто не влияет, в том числе и Scrum Master.
- Для разработки текущего продукта команда полностью обладает всеми навыками - кросс-функциональна.
- В Development Team есть только понятие «Разработчик продукта» – несмотря на то, чем занимается каждый конкретный человек, все должности исключены.
- Development не имеет иерархии или подотделов. Всё решается внутри команды.
- Член команды разработки имеет специализированные знания.
- Каждый член команды разработки имеет специализированные знания, но ответственность лежит на всей команде в целом.



Звездная карта кросс-функциональной команды



Какое место может занимать аналитик в Scrum team?



Аналитик как Product Owner

Аналитик как proxy Product Owner

Аналитик как участник команды

Аналитик как Scrum Master

Возможные функции аналитика в Scrum

Аналитик как PO:

- взаимодействие с заказчиками;
- формирование видения продукта;
- составление гипотез;
- отслеживание ситуации на рынке и позиции конкурентов;
- оценка показателей Продукта;
- etc.



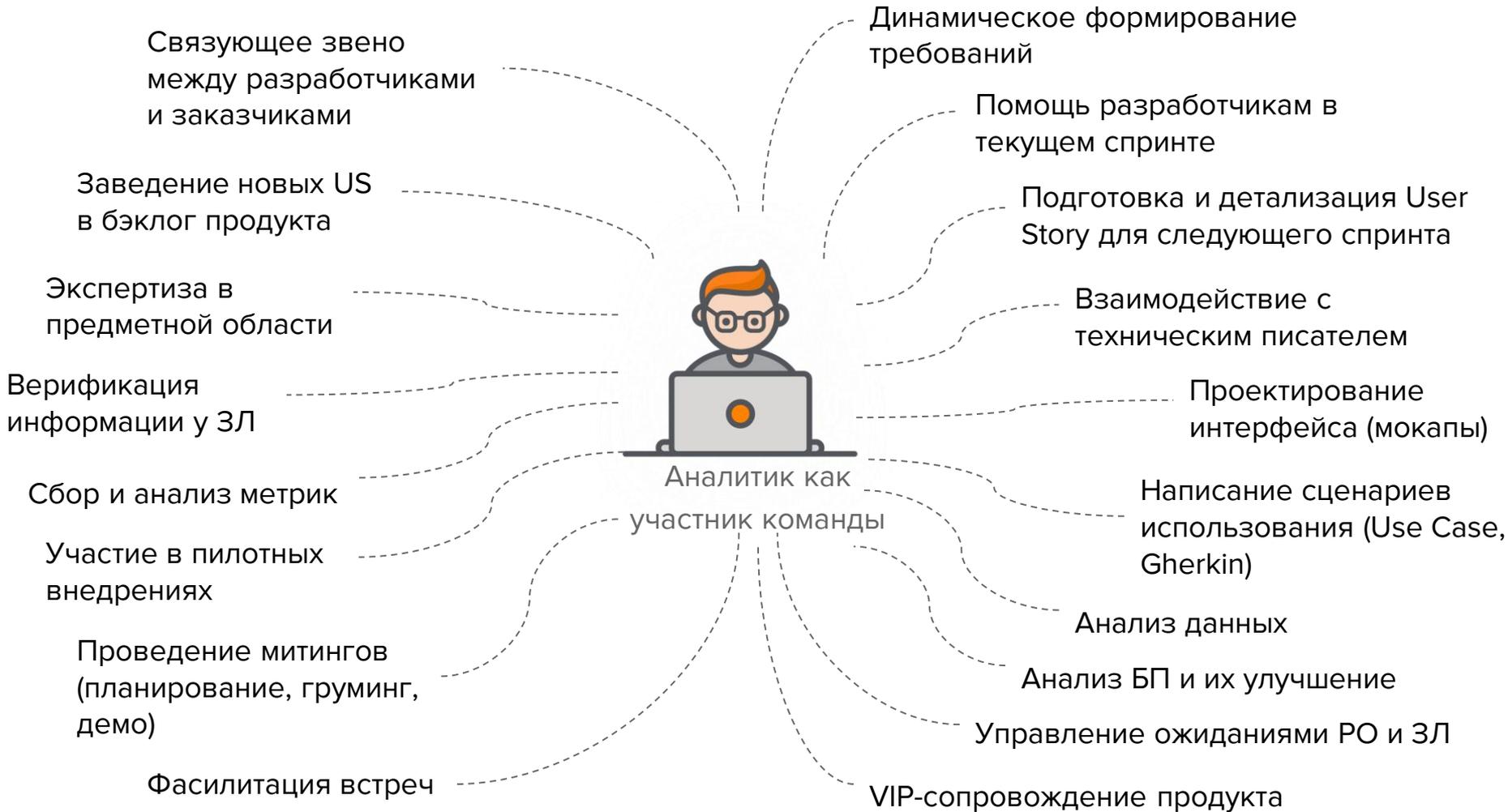
Аналитик как SM:

- проведение ретроспектив;
- фасилитация командных встреч;
- актуализация статуса задач;
- сбор метрик (Velocity);
- личный коучинг членов команды разработки;
- коучинг и менторство PO;
- etc.

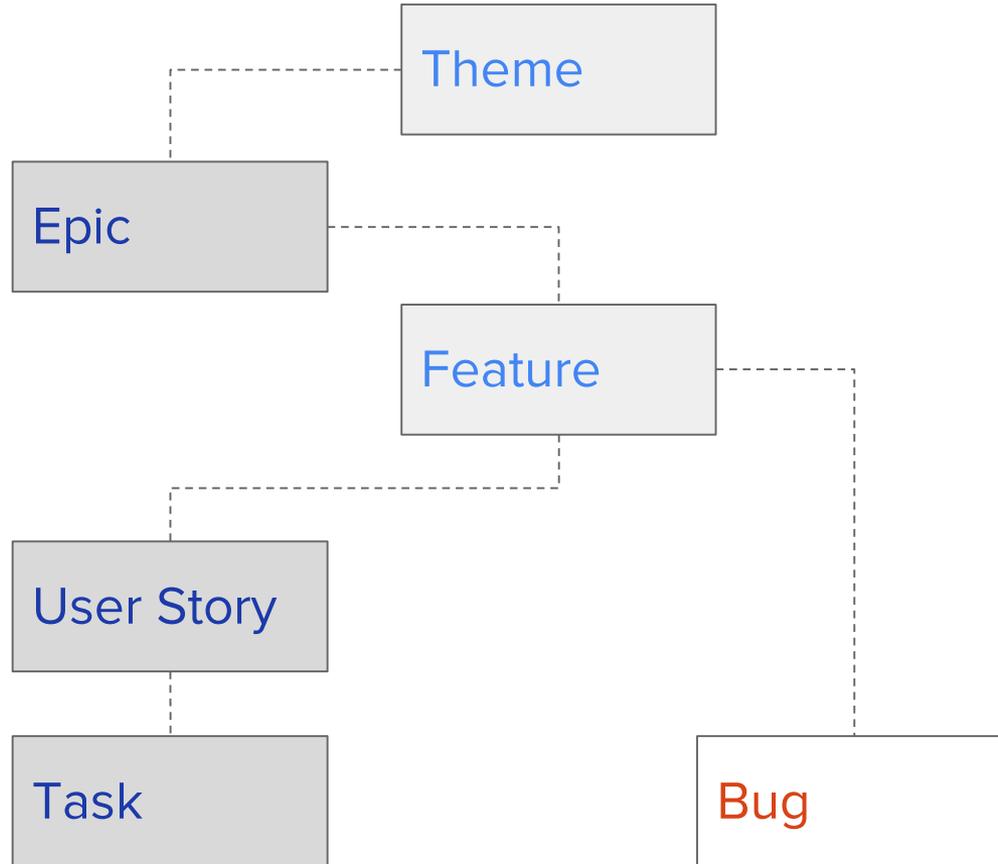
Аналитик как проху PO:

- все то же самое, что и PO, но с его согласованием

Возможные функции аналитика в Scrum



Элементы бэклога



Жизненный цикл User Story



Изменения возможны на любом из этапов.

User Story

Пример:

Как пользователь я могу хранить свои фотографии в системе, чтобы иметь возможность продать их другим пользователям

As a <role>

I want <goal>,

So that <benefit>

Критерии качества User Story (модель INVEST)

I ndependent	С историями легче работать, если они <i>независимы</i> . Хорошо, когда можно планировать и реализовывать их в любом порядке.
N egotiable	Хорошая история отражает суть, а не детали. Со временем User Story может обрастать заметками, тестовыми идеями и т. д. Но эти дополнения не нужны для приоритизации или планирования.
V aluable	История должна нести <i>ценность, пользу</i> для клиента, заказчика, стейкхолдера
E stimable	Хорошую историю можно <i>оценить</i>
S mall	Хорошие истории должны быть <i>маленькими</i> (время на их реализацию не должно превышать длины спринта)
T estable	Хорошая история <i>поддается проверке</i>

The User Story Conversation Canvas

- **Цель:**

для уточнения ожиданий пользователя, для понимания точки зрения «бизнеса»

- **Когда использовать:**

на груминге, планировании и демо

- **Как использовать:**

Использовать готовый шаблон или создать свой и обсуждать по нему US с ЗЛ

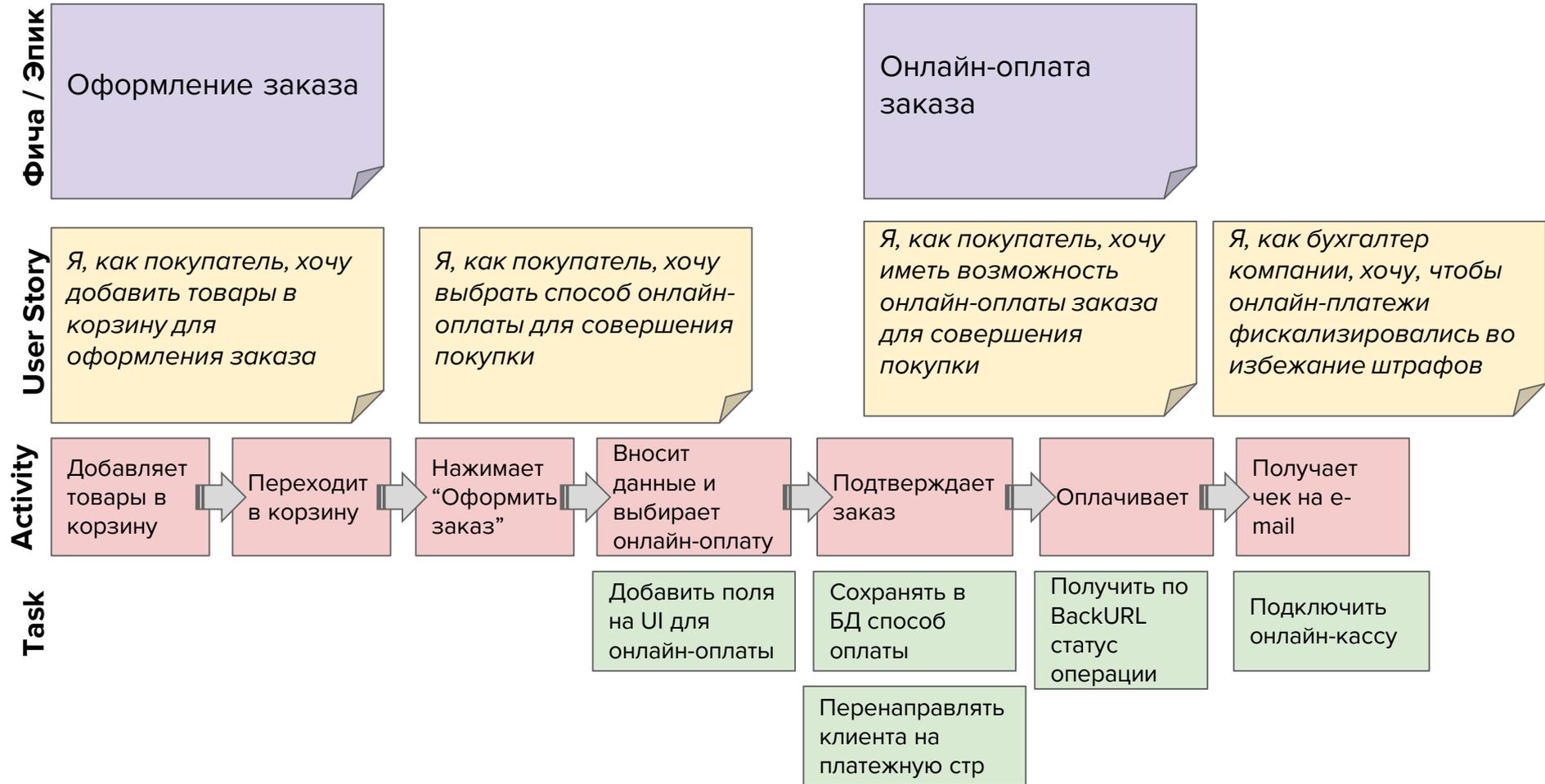
- **Секции шаблона:**

- Персоны
- Пользовательская история (US)
- Критерии приемки
- Контекст
- Критерии готовности взятия US в спринт (DoR)
- Критерии закрытия US (DoD)
- Ожидаемый результат
- Метрики
- Обратная связь
- etc.

The User Story Conversation Canvas

2. User Story As a <User Type> I Want <Do Something> For <Purpose>	1. Personas Personas Users or End Consumers Stakeholders Sponsors Development Team Others (Legal, externals, etc.) Support Team	4. Context <ul style="list-style-type: none">• Source Need• User Sceneries• Business Domain• Business Rules• [Source Epic]• Scope• Hypothesis• Dependencies• Constraints	
3. Acceptance Criteria <ul style="list-style-type: none"><input type="checkbox"/> Acceptance Criteria 1<input type="checkbox"/> Acceptance Criteria 2<input type="checkbox"/> Acceptance Criteria 3•••<input type="checkbox"/> Acceptance Criteria N		5. Definition of Ready What must be fulfilled so that the Story can be built?	6. Definition of Done Conditions of Done specific to the Story
7. Expect Results Expected Solution What do you want to accomplish with the Story? (Example, comply with regulation) Value to earn	8. Metrics <ul style="list-style-type: none">• Points• Effort• Sprint Number• Earned Value (ROI)• Quality• Progress• Number of Users	9. Feedback Level or Grade of People Satisfaction Level of user happiness Generated Impact	

User Story mapping





10/10/00
10/11/00
10/12/00

10/13/00
10/14/00
10/15/00

10/16/00
10/17/00
10/18/00

10/19/00
10/20/00
10/21/00

10/22/00
10/23/00
10/24/00

10/25/00
10/26/00
10/27/00

10/28/00
10/29/00
10/30/00

10/31/00
11/01/00
11/02/00

11/03/00
11/04/00
11/05/00

11/06/00
11/07/00
11/08/00

11/09/00
11/10/00
11/11/00

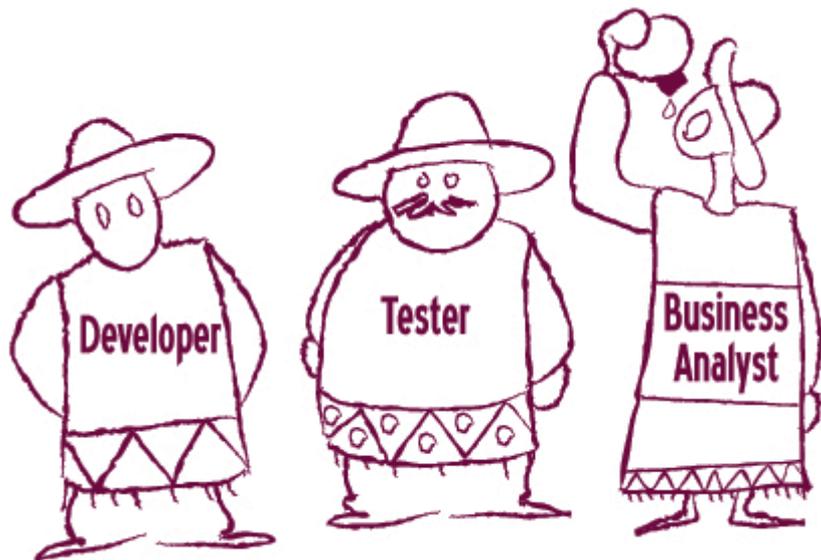
11/12/00
11/13/00
11/14/00

Release
MID-M

Release
10/10/00

Three Amigos practice

Цель: обеспечить правильное понимание потребностей пользователя и предоставить правильное разработанное решение



Как:

- **Аналитик** пишет в блокноте Executable specifications (сценарии использования на Gherkin)
- **Тестировщик**, используя Cucumber (фреймворков для автоматизации тестирования с использованием BDD-подхода), разрабатывает автотесты на основе сценариев
- **Разработчик** определяет зависимости, проектирует архитектуру и пишет код с учетом тестовых сценариев

Gherkin

Gherkin - это предметно-ориентированный язык для написания критериев приемки, имеющий пять ключевых слов:

1. **Scenario** - метка для обозначения сценария, которое вы собираетесь описать
2. **Given** - начальное состояние сценария
3. **When** - конкретное действие, которое пользователь выполняет
4. **Then** - проверяемый результат, обычно вызванный действием в When
5. **And** - это продолжает любой из трех других операторов

Feature & Scenario

Функция (Feature): Короткое, но исчерпывающее описание требуемого функционала

Для того, чтобы достичь определенных целей

В качестве определенного участника взаимодействия с системой

Я хочу получить определенную пользу

Сценарий (Scenario): Какая-то определенная бизнес-ситуация

Дано какое-то условие

И ещё одно условие

Когда предпринимается какое-то действие участником

И им делается ещё что-то

И вдобавок он совершил что-то ещё

То получается какой-то проверяемый результат

И что-то ещё случается, что мы можем проверить

Feature & Scenario

Функция: Онлайн-покупка заказа

Я, как покупатель, хочу выбрать способ онлайн-оплаты для совершения покупки

Сценарий: Покупатель оформляет заказ и выбирает способ онлайн-оплаты

Дано Открыта страница с заказом (корзина)

И все товары в корзине есть в наличии

Когда пользователь нажимает кнопку "Оформить"

То ему доступен выбор способа онлайн-оплаты

Выводы

- Как себя чувствует аналитик в Scrum?
- Нужен ли аналитик в Scrum?
- Кому это может зайти?



Спасибо за
внимание!

