

Про Agile, Scrum и командную работу

Проект

Комплекс взаимосвязанных мероприятий, направленных на создание уникального продукта или услуги в условиях временных и ресурсных ограничений

Характеристики проекта

- Имеет цель
- Результат – что-то уникальное
- Ограничен по времени
- Имеет ограниченные ресурсы
- Реализуется специально организованной командой

Проектный треугольник

Графическая модель, которая отображает взаимосвязь времени, денег, объема работы и качества



При неизменном качестве изменение одного из элементов всегда влечет изменение двух других

Методологии управления проектами

Совокупность методов, принципов и подходов, используемых при ведении проекта

Итеративный подход, в котором ключевую роль играют непрерывные релизы и обратная связь от клиентов

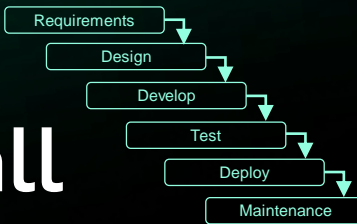


VS



Каскадный подход, в котором работа организована в последовательном порядке и ниспадает каскадом (как водопад)

Waterfall



Линейный и последовательный

Требования фиксируются и определяются заранее

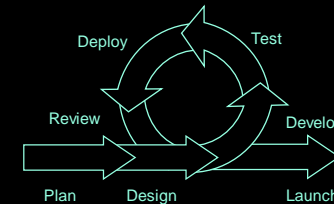
Перед началом проекта готовится комплексная документация

Команды работают изолированно и/или ограниченно до завершения проекта

Участие заказчика до окончательной поставки ограничено

Риски рассматриваются только на этапе инициирования проекта

Agile



Итеративный и инкрементальный

Требования могут меняться в ходе разработки

Документации меньше и акцент делается на создание работающего продукта

Кросс-функциональные команды тесно сотрудничают в ходе проекта

Заказчик вовлекается на каждом этапе разработки

Риски управляются на протяжении всего цикла разработки

Подход

Требования

Документация

Работа в команде

Взаимодействие с заказчиком

Риски

Проектный треугольник

для разных методологий

Waterfall

Фиксируем

Объем работ

Качество

Стоимость

Сроки

Оцениваем

Agile

Стоимость

Сроки

Качество

Объем работ

Гибкие методологии разработки

Agile

система ценностей (философия), которая объединяет все гибкие методологии разработки

Ценности Agile
родились в 2001 в
Agile-манифесте в
результате обобщения
многих существующих
на тот момент
методологий
разработки их
авторами

Люди и их
взаимодействие

важнее
процессов и
инструментов



Работающий
продукт

важнее
исчерпывающей
документации



Ценности
Agile

Сотрудничество
с заказчиком

важнее
согласования
условий контракта



Готовность
к изменениям

важнее
следования
первоначальному плану



Agile фокусируется на быстрой разработке и поставке новых продуктов, максимально отвечающих потребностям бизнеса

Scrum

Фреймворк по организации итеративно-инкрементальной разработки, который помогает поставлять продукт быстрее и с более высокой ценностью для клиента



Роли

Product Owner

Несет ответственность за максимизацию ценности продукта и эффективное управление Product Backlog

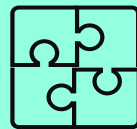
Scrum Master

Несет ответственность за применение Scrum в соответствии с Руководством по Scrum

Developers

Несут ответственность за создание Sprint Backlog и достижение Sprint Goal

Product Owner, Scrum Master и Developers составляют Scrum Team



Артефакты

Product Backlog

Упорядоченный и постоянно обновляемый список того, что необходимо улучшить для продукта

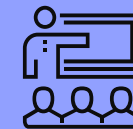
Приверженность Product Goal (будущему состоянию продукта, которое выступает в качестве конечной цели)

Бэклог спринта

Список рабочих задач, пользовательских историй или багов, отобранных для спринта

Инкремент

Готовый к использованию конечный продукт по итогам спринта



Церемонии

Sprint Planning

Планирование работы для выполнения Developers в течение Sprint
Приверженность Sprint Goal – единственной цели на Sprint

Daily Scrum

Инспекция прогресса в достижении Sprint Goal (15-минутное событие для Developers)

Sprint Review

Инспекция результата Sprint и выявление возможностей для адаптации

Sprint Retrospective

Инспекция того, как прошел Sprint (что было хорошо и с какими проблемами столкнулись)

Scrum

Графическое представление фреймворка



Правила работы в командах

Команда



Владелец продукта
(Product owner)



Аналитик
(Business Analyst)



Дизайнер



Разработчик
(Developer)



Маркетолог
(Marketer)

Артефакты

Приоритизированный бэклог
(доски визуализации)

Документация по проекту

Описание задач (User Story)

MVP

События

Sprint Planning

Daily Scrum

Sprint Review

Sprint Retrospective

Вопросы?

Дополнительные материалы

Product Backlog Refinement

(Уточнение бэклога продукта)*



Провести ревизию пользовательских историй в бэклоге продукта: написать новые, оценить их и удалить неактуальные, декомпозировать, упорядочить истории. Определить задачи готовые к разработке в ближайшем спринте

Обсудить задачи и определить необходимость их декомпозиции, упорядочивание задач и их оценка командой.



1 раз в спринт



1-2 часа



Делает Владелец продукта при участии всей Scrum-команды

*не относится к официальным мероприятиям Scrum.

Чек-лист Product Backlog Refinement



Обязательно:

- Есть как минимум 1 встреча по актуализации Бэклога продукта в календаре Команды
- На встрече присутствуют все необходимые и все заинтересованные члены Команды
- На встречу ВП приносит новые и/или еще непроработанные Командой элементы Бэклога Продукта, которые ВП планирует реализовывать в ближайшие 1-2 спринта
- ВП озвучивает каждую задачу (элемент Бэклога (PBI)), команда с ВП уточняют детали
- Команда совместно с ВП формирует критерии приемки по каждому элементу Бэклога Продукта
- Вопросы, на которые не получилось ответить в рамках встречи фиксируются и откладываются до следующей встречи. В Команде есть понимание, кто берет на себя ответственность на то, чтобы найти ответы на них
- Команда оценивает задачу (или задачи) любым удобным ей способом (например, с помощью "Покер планирования")
- Команда декомпозирует задачи на более маленькие, каждая из которых должна нести ценность конечному клиенту
- Команда и ВП по итогам актуализации задач совместно проверяют их на критерии готовности для взятия в Спринт (DoR)



Желательно:

- По результатам встречи часть старых и неактуальных для Продукта задачи удаляется из Бэклога Продукта
- Если оценка задачи будет больше $\frac{1}{2}$ - $\frac{2}{3}$ от исторической производительности Команды (velocity) в прошлых Спринтах, команда такие задачи декомпозирует
- ВП и Команда общается на встрече с пользователями, чтобы понять, какие потребности необходимо закрыть
- Встреча проводится с заданной и стабильной регулярностью
- Результаты декомпозиции и проработки задач фиксируются в JIRA
- Все участники встречи проявляют на ней активность

Sprint Planning (Планирование спринта)



Определить Цель спринта и сформировать бэклог спринта. Сформировать план выполнения цели



1 раз в спринт, в 1-ый день спринта.



для спринта длиной в:

- 1 месяц - не более 8 часов
- 2 недели - не более 2-х часов
- 1 недели - не более 1-ого часа



Вся Scrum-команда

*Состав спринта не изменяется, но может быть отмен по инициативе Владельца продукта.

Чек-лист Sprint Planning



Обязательно:

- Есть встреча по Планированию Спринта в календаре Команды
- На встрече присутствуют все члены Команды
- До начала планирования незавершенные задачи предыдущего спринта возвращены в Бэклог Продукта
- На встречу ВП приносит Бэклог Продукта, приоритизированный по ценности (в JIRA важные для ВП задачи находят сверху)
- ВП озвучивает каждую историю, команда с ВП уточняют незакрытые по задаче вопросы
- Если задачу не успели ранее разобрать (на актуализации Бэклога), то задача попадает в Спринт, если Команда готова дать прогноз, что сможет сделать задачу
- Команда совместно с ВП формирует Цель Спринта. Задачи, относящиеся к Цели Спринта расположены сверху Бэклога Спринта.
- Команда и ВП до старта Спринта совместно проверяют задачи в Бэклоге Спринта на критерии готовности для взятия в Спринт (DoR)
- Команда и ВП при формировании Бэклога Спринта ориентируются на историческую производительность Команды (velocity) в прошлых Спринтах
- Команда и ВП при формировании Бэклога Спринта ориентируются дополнительно на доступность членов Команды (плановый отпуск, командировки, болезни и проч)
- Команда декомпозирует задачи на подзадачи (в этот момент ВП может уйти со встречи)
- ВП не давит на команду взять больше, чем может Команда

Желательно:

- В Бэклоге Спринта находится 5-10 элементов (например, Пользовательских Историй (то есть не 1-2) для снижения рисков
- Цель спринта не "Сделать все задачи", а отражает определенный измеримый результат, отражающийся на пользователях. Грешим иногда тем, что не описываем цель спринта
- Календарь работ на спринт



Daily Scrum (Ежедневный митинг)



Инспекция прогресса в достижении Цели спринта. Синхронизация работ, планирование на день, выявление проблем и степень влияния на Цель спринта.



Ежедневно.



Не более 15 минут. Обычно в начале рабочего дня.



Команда разработки+ Scrum-мастер. Опционально Владелец Продукта.

Чек-лист Daily Scrum



Обязательно:

- Есть повторяющаяся встреча в календаре Команды
- Встреча начинается вовремя. На встрече присутствуют все члены Команды (ВП может отсутствовать)
- Встреча проходит с выведенным Бэклогом Спринта в JIRA или у физической доски
- Члены Команды кратко рассказывают о прогрессе и планах достижения Цели Спринта, а так же обозначают трудности к ее достижению
- Это встреча Команды, поэтому нет отчета перед ВП или СМ (члены Команды обращаются друг к другу , а не только к ВП/СМ)
- Команда укладывается в отведенные 15 минут
- СМ паркует (фиксирует) проблемы, которые есть у Команды
- Отдельные Члены Команды встречаются для решения выявленных проблем после встречи



Желательно:

- Последовательностью выступления членов Команды управляют члены Команды, а не СМ/ВП

Sprint Review (Обзор спринта)



Оценка выполнения цели и результата спринта, демонстрация продукта и, по необходимости, адаптация бэклога продукта.



1 раз в конце каждого спринта



Для спринта длительностью:

- 1 месяц - не более 4 часов
- 2 недели – 2 часа
- 1 неделя – 1 час



Вся Scrum-команда и все заинтересованные лица (в т.ч. Заказчик)

Чек-лист Sprint Review



Обязательно:

- Есть встреча в календаре Команды
- На встрече присутствуют все члены Команды
- На встречу приходят пользователи и другие заинтересованные лица (то есть это не встреча, на которой сидят только члены Команды)
- СМ в начале встречи объявляет о ее цели (получить ОС по Продукту), о правилах работы на встрече и повестке
- Команда демонстрирует новый функционал
- СМ организует процесс сбора ОС таким образом, чтобы каждый смог ее дать
- Команда и ВП фиксируют всю полученную обратную связь
- ВП объявляет собравшимся о дальнейших планах развития Продукта (возможно, при этом демонстрирует Бэклог Продукта)
- СМ управляет дискуссией, помогает со сложными (агрессивный, слишком активный, молчаливый) участниками встречи
- Встреча неформальная. Не является отчетом о проделанной работе



Желательно:

- ВП озвучивает Цель Спринта и планируемые в прошлом Спринте задачи
- ВП честно озвучивает, что получилось сделать, а что нет (не скрывает факта не сбывшихся прогнозов, которые дала Команда)
- Команда дает пользователям попробовать новый функционал

Sprint Retrospective (Ретроспектива)



Провести анализ спринта в отношении людей, взаимоотношений, процессов и инструментов работы команды в спринте, и создать план улучшений командной работы в следующем спринте.



1 раз в завершении каждого спринта после проведения Обзора спринта, перед Планированием следующего спринта.



Для спринта длительностью в:

- 1 месяц – не более 3 часов
- 1-2 недели – 1-2 часа



Вся Scrum-команда

*Ретроспектива – это закрытая церемония команды.

Чек-лист Sprint Retrospective



Обязательно:

- Есть встреча в календаре Команды
- На встрече присутствуют все члены Команды (в том числе и ВП)
- СМ озвучил цель встречи (повышение эффективности команды), напомнил про командные договоренности (правила общения), и озвучил повестку
- СМ подготовил дизайн: есть разминка (легкая, веселая активность, направленная на фокусирование Команды на цель встречи)
- Делается разбор результатов прошлой ретроспективы, в частности озвучивает action item'ы (AI) с прошлой ретроспективы. Если AI не выполнены, то для актуальные AI (и причины их невыполнения) добавляются в повестку ретро
- СМ организовал процесс по сбору данных
- СМ организовал процесс по генерации идей (по выявленным проблемам Команды)
- СМ организовал процесс по выработке решений (когда понятно, что именно будет сделано) и взятию членами Команды на себя ответственности (кто именно будет реализовывать AI)
- Взятые AI появляются в Бэклоге Продукте (в JIRA)
- Команда запланировала часть AI в следующий спринт



Желательно:

- СМ принес красиво оформленные флипчарты и другие элементы, которые создают правильное настроение на встрече
- СМ не повторяется в дизайнах, ищет новые формы работы на встрече, чтобы не было скучно
- СМ дает ОС Команде по тому, что, по его мнению произошло (с его точки зрения) в прошлом Спринте, не дает замалчивать конфликты, сложности, о которых Команда не хочет говорить
- Задачи, которые не входят в зону контроля и/или влияния у Команды, эскалируются на уровень Продукта/Кластера или на уровень Трейба

User Story

Для чего применяется User Story?

- Для описания элементов бэклога
- Для лучшего понимания пользователей
- Для описания требований к продукту на понятном для всех языке

Как формулировать User Story?

User Story — это ответы на 3 вопроса, связанные в одно предложение:

- Что это за пользователь?
- Какое действие он хочет выполнить в продукте или какой результат от продукта хочет получить?
- Зачем это ему?



Пользовательские истории:

Как **<роль или тип пользователя>**,
я хочу/могу **<выполнить действие или получить результат>**,
чтобы **<получить ценность>**

User Story

Пример:



Как **"новый сотрудник компании"**,
я хочу **"узнать что такое Scrum"**,
чтобы **"лучше работать в команде"**

Хорошая пользовательская история

INVEST — критерий хорошей истории:

Independent — независимая от других историй, то есть истории могут быть реализованы в любом порядке

Negotiable — обсуждаемая, отражает суть, а не детали; не содержит конкретных шагов реализации

Valuable — ценная для клиентов, бизнеса и стейкхолдеров

Estimable — оцениваемая по сложности и трудозатратам

Small — компактная, может быть сделана командой за одну итерацию

Testable — тестируемая, имеет критерии приемки

Эти критерии не всегда достижимы, но чем больше историй будут им удовлетворять, тем более гибким будет ваш процесс разработки продукта.

User Story Mapping

Для чего применяется USM?

- Для проектирования пользовательского опыта в продукте
- Для определения границ MVP (минимальной работоспособной версии продукта) и планирования релизов на базе пользовательского сценария
- Для формирования единого понимания пользователя у команды разработки и заинтересованных лиц

Как построить User Story Mapping?

Для построения USM вам потребуется:

- Инструмент визуализации: стикеры или электронный инструмент вроде Miro или Mural
- Владелец продукта и команда разработки
- Представление о пользовательском сценарии (если у вас уже есть построенный CJM – берите его за основу)
- 1-2 часа

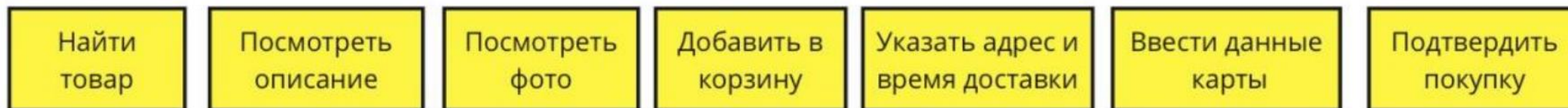
Ваша задача — спроектировать по шагам действия пользователя в продукте на основе его реального сценария.

User Story Mapping_Пример

Магазин цветов решил запустить сайт.

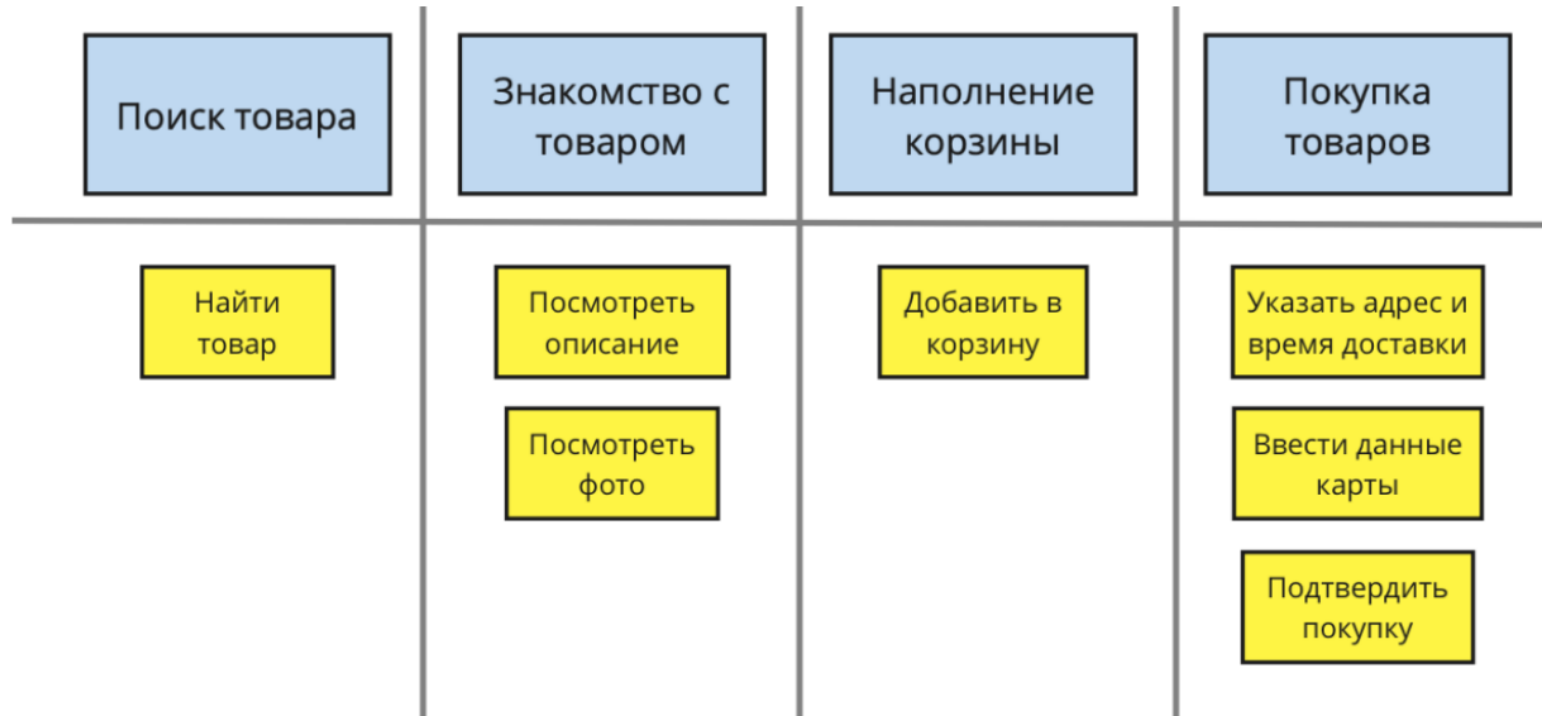
Визуализируем опыт клиентов с помощью техники USM.

1. Расскажите историю клиента по шагам



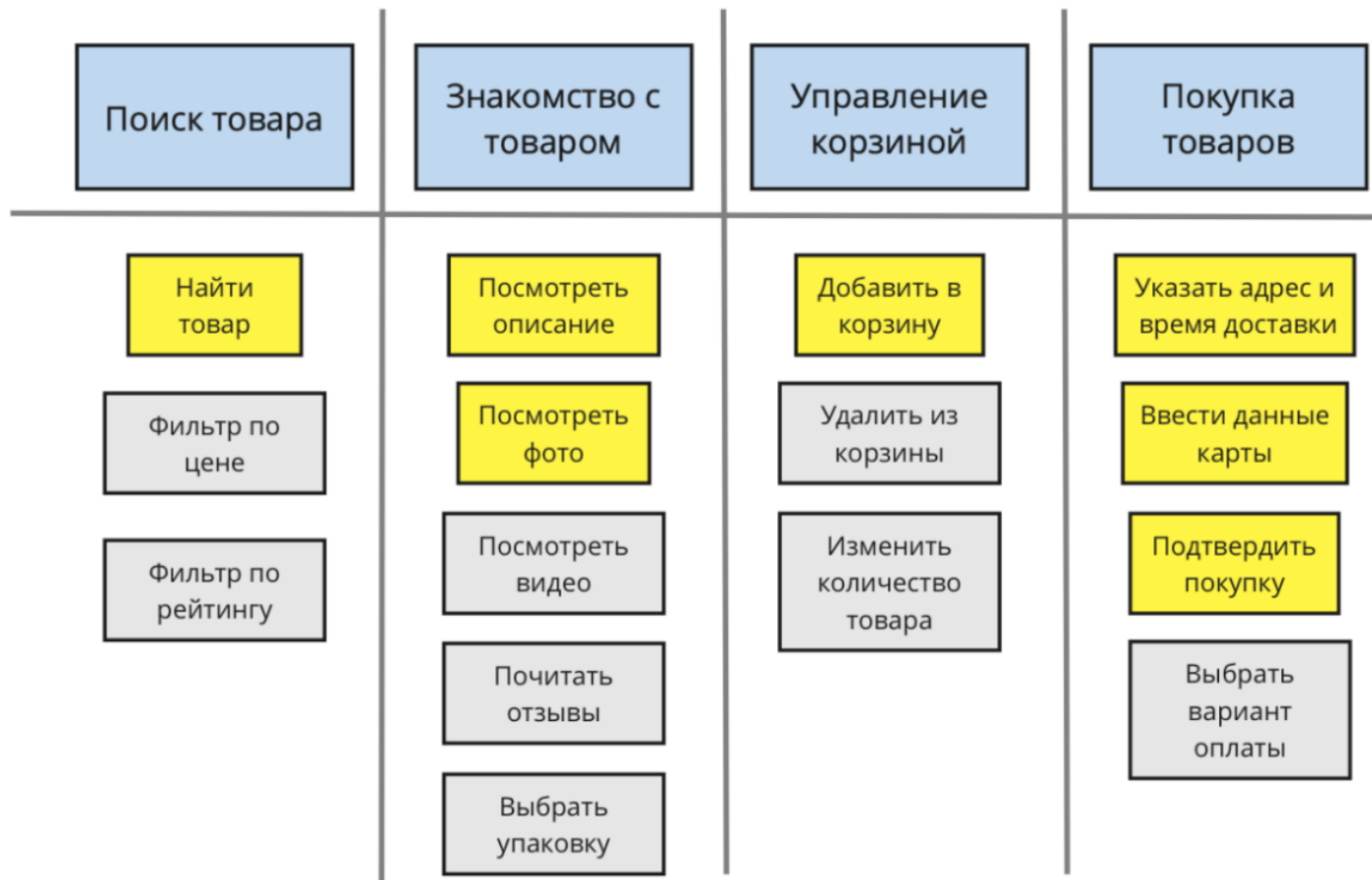
User Story Mapping_Пример

2. Сгруппируйте действия клиента в этапы



User Story Mapping_Пример

3. Заполните пробелы в истории (User Story)



User Story Mapping_Пример

4. Приоритизируйте истории внутри каждого этапа пути

	Поиск товара	Знакомство с товаром	Управление корзиной	Покупка товаров
Сделать обязательно	Найти товар	Посмотреть описание Посмотреть фото	Добавить в корзину Удалить из корзины	Указать адрес и время доставки Ввести данные карты Подтвердить покупку
Хорошо бы сделать	Фильтр по цене	Выбрать упаковку Посмотреть видео	Изменить количество товара	Выбрать вариант оплаты
Можно будет сделать	Фильтр по рейтингу	Почитать отзывы		

User Story Mapping_Пример

5. Выделите релизы

