

Пользовательские истории для Google Keep

Задание #3 по курсу “Прикладной системный анализ III (Введение в бизнес-анализ)”

Преподаватель: Атрохов К. Г.

Весна 2021

Задание

В этом задании вам нужно сделать следующее:

- Изучить сервис для создания заметок [Google Keep](#)
- Определить пользовательские сценарии и реализующие их фичи (возможности) и описать их в виде историй (User stories) с разбиением на эпика (Epics)
- Расставить фичи по приоритетам и отобрать те из них, которые вы бы поставили в первый публичный релиз (Phase 1)
- Создать онлайн канбан доску [Trello](#) с 4 дорожками:
 - **Backlog** -- сюда добавить все юзер стори, расставить их по приоритетам, окрасить в цвета в соответствии с эпиками
 - **To do** -- сюда перенести стори из MVP и распределить их по участникам команды
 - **In progress** и **Done** -- каждый участник команды подробно описывает назначенные ему стори (добавляет критерии приемки и мокапы) и перетаскивает их в сначала в In progress, а потом в Done

Для выполнения задания вам необходимо использовать [канбан-доску](#) Trello. Это довольно простой сервис, почитать про который можно [здесь](#), [тут](#) и [там](#).

Выделение пользовательских историй

Пользовательские истории (User Stories) наравне с вариантами использования (Use Cases) являются основными способами описания требований. Истории лучше подходят командам с опытными программистами: для тех не нужно писать слишком подробные описания задач. Также истории хорошо подходят agile-командам в начале развития проекта -- набросать список историй гораздо легче, чем расписать все юзкейсы.

Действительно, история может состоять из одного предложения типа: я, как <пользователь>, хочу иметь <возможность>, чтобы <цель/причина>. Такие истории легко создавать (например, идея пришла в голову или подсмотрена у конкурента).

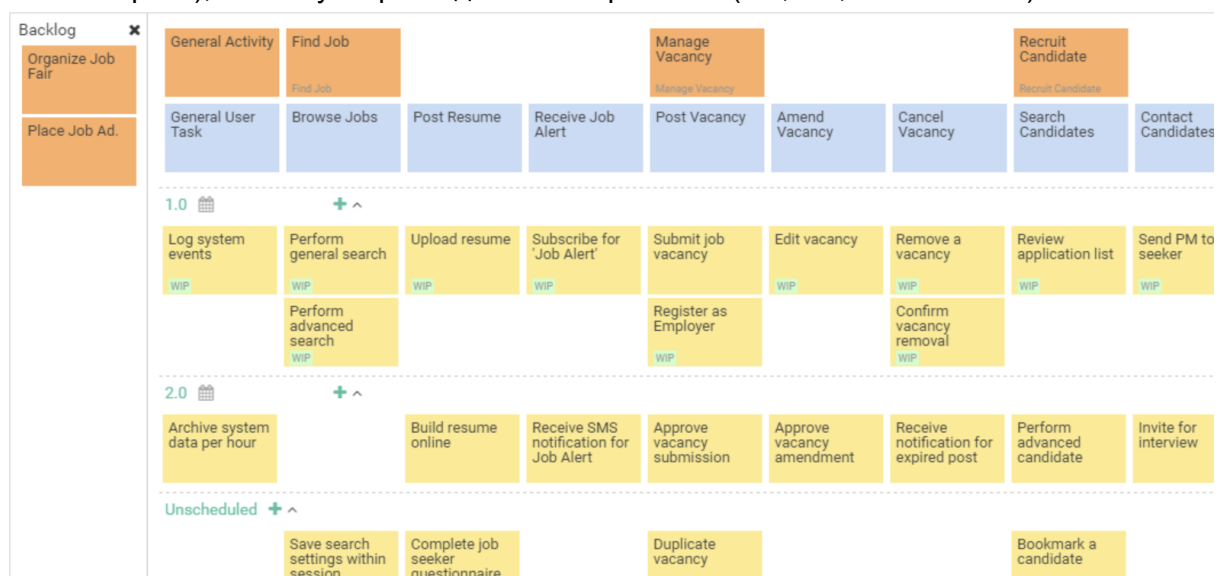
Например, при создании почтового веб-клиента сразу приходит на ум много историй:

- Я, как пользователь, хочу иметь возможность создавать и отправлять письма (здесь даже “потому что” не нужно)
- Я, как пользователь, хочу иметь возможность прикрепить к письму файлы разных форматов, потому что часто одного текста недостаточно или адресат требует именно файл
- Я, как пользователь, хочу иметь возможность не отправлять письмо сразу, а сохранить его как черновик, потому что порой ты не можешь написать текст сразу или тебя может что-то отвлечь
- Я, как пользователь, хочу иметь возможность форматировать текст письма, например, делать текст жирным, менять его цвет, создавать списки и т.п., так как это сделать мое письмо более читабельным и т. д.

Создать историю очень легко, но реализовать ее может быть достаточно сложно и долго. Слишком большую историю (по оценке команды) разбивают на несколько более простых. Так получается эпик (Epic) -- группа историй, которые в совокупности ведут к ощутимому и важному результату.

Многие команды по-прежнему записывают истории на бумажные карточки и клеят их на доску или стену.

- Слева бэклог -- хранилище задач с приоритетами
- Отобранные большие задачи собираются вверху (оранжевые карточки на скриншоте)
- Они разбиваются на задачи поменьше (голубые карточки)
- Внизу желтые карточки с задачами, которые можно выполнить за короткое время (в идеале за 1 день, чтобы отчитаться на завтрашнем скраме)
- Как вы видите, желтых задач может быть много (чтобы выполнить за один спринт), поэтому их раскидывают по релизам (1.0, 2.0, Unscheduled)



Существует несколько методов декомпозиции задач:

- *Пройтись по [CRUD](#)* -- история “Как пользователь я хочу управлять своим аккаунтом, чтобы видеть и иметь возможность обновить настройки” превращается в задачи по созданию, просмотру, обновлению и удалению аккаунта
- *Пройтись по жизненному циклу актора, сущности или процесса* -- история “Я как менеджер по контенту хочу иметь возможность создавать новости на сайте компании, чтобы клиенты и медиа могли их прочитать” может трансформироваться в задачи “создать новость, после чего та отправится на одобрение редактором”, “одобрить публикацию”, “отменить публикацию с комментариями” и т.д.
- *Учесть позитивные и негативные исходы* -- как мы сделали в прошлом примере с одобрением и отменой
- *Разобрать возможные бизнес-кейсы* -- анализируя историю “Как путешественник, я хочу иметь возможность искать полеты с гибкими датами, чтобы легче планировать свой отпуск” с владельцем продукта (или экспертом) можно прийти к разному способу задания гибких дат, например, “выходные в июле” или “дата плюс-минус несколько дней”, а это разные задачи
- *Разобрать на части учитывая возможные ветвления* -- например, сохраняем в файл => отдельная задача под каждый формат, в системе разные типы доступа к информации => отдельная задача под каждый тип и т.д.

Чтобы взять историю в работу ее нужно дополнить критериями приемки (Acceptance Criteria). По сути это и есть описание функционала, но в виде чеклиста, что может быть удобным и для программиста, и для тестировщика.

Например, добавим критерии приемки в историю про прикрепление файлов для почтового клиента:

- Прикрепление файлов опционально, письмо можно отправить без них
- Размер прикрепляемого файла ограничен и не должен превышать 25 МБ
- Прикрепляемый файл загружается в облако, прогресс загрузки отображается пользователю и письмо не может быть отправлено во время этого процесса
- Пользователь должен видеть все прикрепленные файлы до отправки письма
- Если в процессе прикрепления произошла ошибка, то это также должно быть отображено
- В списке писем те из них, которые имеют прикрепленные файлы, должны быть помечены специальной иконкой

Критерии приемки уточняются вместе с командой (программистами, дизайнерами, тестировщиками). Если для задачи требуется интерфейс, то его прикладывают в виде мокапов или готового дизайна.

Подробнее про пользовательские истории можно прочитать [здесь](#) (слайды [доступны по ссылке](#)). Также хорошие слайды [смотрите здесь](#), а почитать про разбиение скоупа задач (Scope) на истории -- [здесь](#).

Приоритезация историй

Открывая какой-нибудь сервис, вы видите его актуальную версию, которая, возможно, создавалась годами. Так, сервис Кеер был запущен в 2013 году. Никто не пытался реализовать все фичи сервиса сразу -- для такого подхода даже у Google не хватит ресурсов. Поэтому при разработке ПО все потенциальные фичи реализуются согласно приоритетам. Приоритезированный список фич (историй) называется бэклогом (Backlog).

В качестве метода приоритезации вы можете выбрать наиболее популярный качественный метод: по усилиям (Effort) / ценности (Value)

Легкие и ценные (High)	Трудные и ценные (Medium)
Легкие и не очень ценные (Medium)	Трудные и не очень ценные (Low)

Замечание: Вам будет сложно оценить сложность задач без участия программистов. Несколько более простых в обращении методов приоритезации вы найдете в [презентации Майка Кона](#).