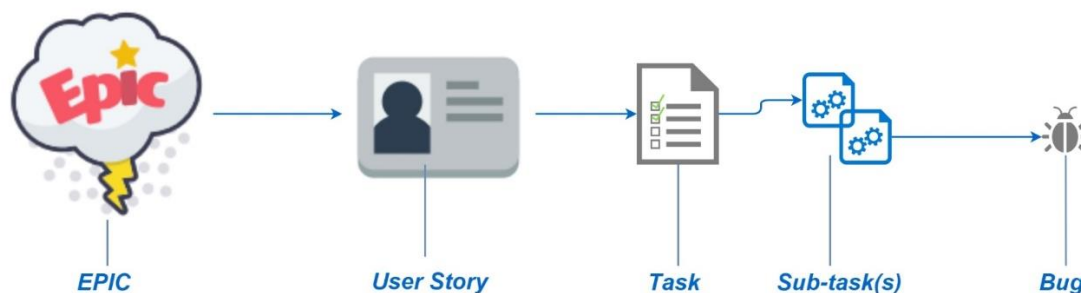


Что такое артефакт проекта и зачем он нужен?

Артефакт – это искусственно созданный объект, который наделяется каким-то определенным смыслом тем, кто его создал. Например, схема, документ, прототип – это все артефакты. Артефакт это сжатое, емкое понятие единственная цель которого – оптимизация/минимизация времени на объяснение сути объекта, оптимизация текста, затрачиваемого на описание чего-то и так далее.

Для процесса разработки программного обеспечения был выведен следующий общий набор артефактов:



EPIC (Эпик)

Эпик – это большой компонент разрабатываемой программы. В каждом проекте есть несколько эпиков, которые показывают, что мы ожидаем по окончании проекта.

Пример:

Проект: «Мобильное приложение UBER».

Эпики:

«Система регистрации», «Интерактивная карта», «Платежная система», «Рейтинговая система» и так далее.

Типовое описание эпика:

Система регистрации – необходимо обеспечить возможность регистрации посредством социальных сетей, с обязательной верификацией номера мобильного телефона.

Пример:

Система регистрации. Целевая аудитория состоит из мужчин и женщин в пропорции (65% / 35%). Возраст от 18 до 45 лет. Ввиду того, что в стратегических планах компании присутствует расширение и запуск в Китае, необходимо учесть, что Facebook там запрещен, но присутствуют локальные соц. сети Weibo и Wechat. Для аналитики нам потребуется собирать информацию которая помогла бы выявить специфику в группе пользователей, а также облегчить процесс регистрации для пользователей целевой аудитории.

User Story (Пользовательская история)

Данный артефакт представляет требования к разрабатываемому компоненту.

Один Эпик может содержать в себе десятки пользовательских историй.

В отличие от Эпика, в Историях находится именно та информация, которая напрямую необходима программисту для разработки компонента.

Эпик «Платежная Система» может содержать следующие Истории:

«Страница платежных методов», «Прикрепление банковской карты», «Прикрепление PayPal аккаунта», «Страница оформления заказа» и далее.

Пример пользовательской истории «Страница платежных методов»:

Как пользователь UBER, я бы хотел иметь возможность доступа к странице «Платежные Методы» из левого меню. На этой странице я бы хотел видеть две кнопки:

- Добавить новую карту
- Добавить PayPal аккаунт

Если у меня уже есть добавленные методы платежа, я бы хотел видеть их в списке, с указанной датой добавления и кнопкой удаления платежного метода, которая, при нажатии, открывала бы окно подтверждения действия с вариантами «Да/Нет».

Любая деталь пользовательской истории важна. Деталь, которая будет упущена – это минус несколько минут/часов/десятков часов для программистов работающих над реализации истории.

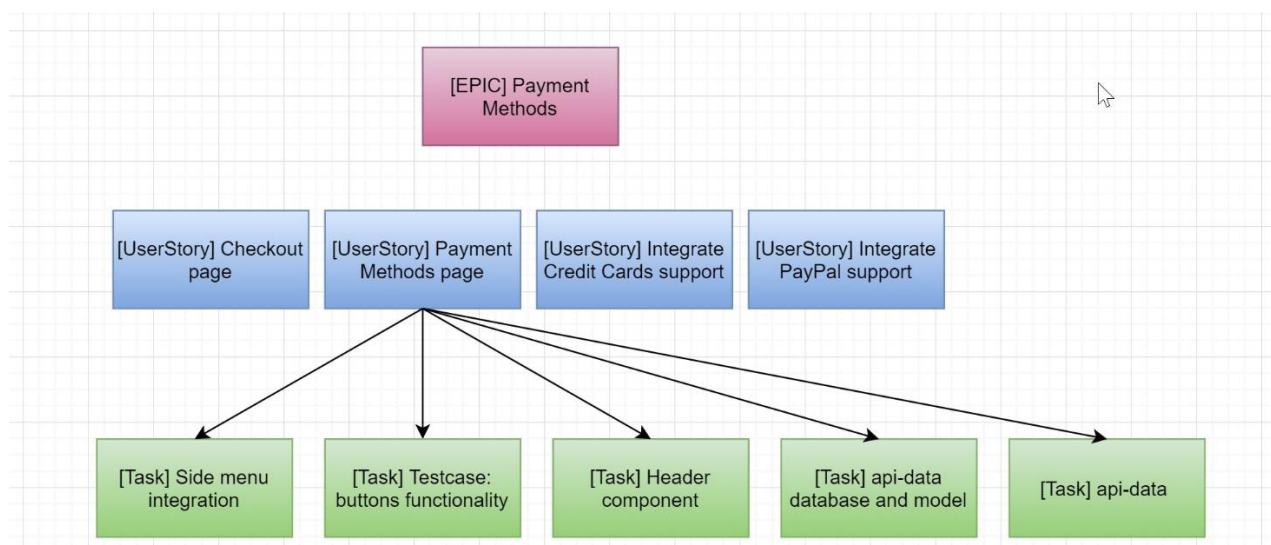
Для написания идеальной пользовательской истории, нужно понять:

- Как этот компонент системы должен работать?
- Какие вопросы могут возникнуть у программистов?
- Какие вопросы могут возникнуть у дизайнеров?
- Какие вопросы могут возникнуть у тестировщиков?
- Какие вопросы могут возникнуть у конечных пользователей этого компонента?

Task (Задача)

Пользовательские истории делятся на задачи.

Пример для разработки истории «Страница платежных методов»:



Sub-Task (Подзадача)

Подзадача – это результат деления задачи на более мелкие составляющие.

Bug (Баг)

Ошибка, возникшая в программе, и результат не соответствует описанию того, как должен работать компонент. Например, при нажатии на кнопку «Прикрепить PayPal» в открывшемся меню видим приглашение прикрепить банковскую карту.

Вне зависимости от того, кто обнаружил баг, он должен быть задокументирован, чтобы команда разработчиков его исправила.

Полный текст:

<https://keepsimple.io/ru/project-artifacts-and-their-importance>