

Лабораторная работа № 2

Разработка структуры взаимодействия «пользователь-интерфейс»

Теоретические сведения

Описание взаимодействия «пользователь-интерфейс» осуществляется посредством пользовательских историй (Userstories), сценариев (Scenarios), вариантов использования (Usecases) и Пользовательских историй Agile методологии (Agileuserstories). На сегодняшний день в UX терминологии есть некоторая путаница: разные авторы под Userstories, Usecases, Scenarios, Agileuserstories могут понимать разные инструменты. Но все их можно отнести к категории «пользовательский сценарий».

Пользовательский сценарий (User Scenario) — это повествование, которое позволяет определить, почему пользователи оказываются на сайте/в приложении и как реализуют свои планы с помощью проектируемого продукта. Чаще всего пользовательские сценарии создаются в формате коротких рассказов о некоем эталонном пользователе, цель которого — удовлетворить свою потребность посредством создаваемого вами сайта или приложения.

Сценарии пишутся до начала работ собственно по созданию сайта/приложения, они сфокусированы на потребностях и действиях. Технические аспекты (инструменты), особенности интерфейса в сценарии не рассматриваются.

Пользовательские сценарии помогают

1. понять ключевые потребности и мотивы клиентов;
2. найти самые важные моменты, которые необходимо учесть при разработке взаимодействия «пользователь-интерфейс»;
3. анализировать пользовательский опыт;
4. изучить и лучше понимать предпочтения пользователей;
5. спроектировать интерфейс так, чтобы он соответствовал привычным, типичным для клиентов паттернам и позволял достичь цели, затратив минимум сил и времени;
6. выяснить, на каких этапах процесса взаимодействия с интерфейсом пользователям потребуется дополнительная помощь.

Рассмотрим, что общего и чем отличаются различные виды пользовательских сценариев. Одну из наиболее популярных классификаций пользовательских сценариев разработали Дэвид Бенъон, Сьюзан и Фил Тернеры — авторы книги *Designing Interactive Systems — People, Activities, Contexts, Technologies*. Они выделяют 4 вида сценариев в зависимости от этапа проекта и целей их создания:

1. пользовательские истории,
2. концептуальные сценарии,
3. конкретные сценарии,
4. варианты использования.

Соотношение между этими видами сценариев представлено на рисунке 1.

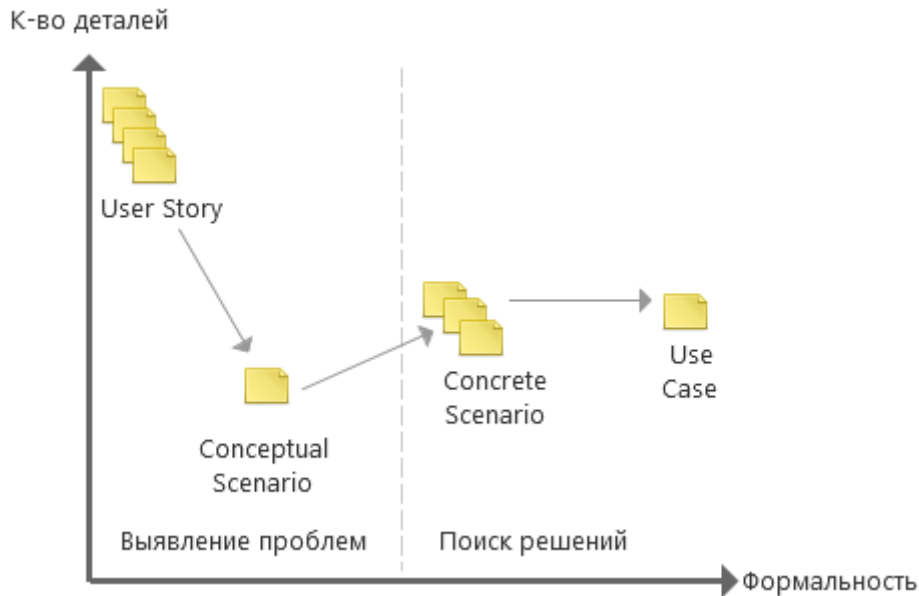


Рис 1. Соотношение между различными видами сценариев[1].

К этому списку целесообразно добавить пользовательские истории Agile методологии (Agileuserstories), предложенные Дж. Паттоном, они занимают промежуточное место между конкретными сценариями (Concrete scenario)и вариантами использования (Use case).

Рассмотрим на примерах различные виды сценариев[1].

Пользовательские истории (User story)

Пользовательские истории (User stories) — краткое содержание «легенды» пользователя и описание его потребностей, связанных с разрабатываемым продуктом. Характеристики пользовательских историй:

- относительно небольшой размер;
- максимальная детализация: содержат подробности о пользователе, его особенностях, мотивах и привычках;
- от лица пользователя описывается потребность, которую помогает удовлетворитьпроектируемая система.

Пользовательские истории описывают потребность пользователя в его обычной повседневной жизни (*Raw user needs*). Эта потребность существует независимо от того, создано для нее программное обеспечение или нет. Пользовательские истории пишутся литературным языком, поэтому легко читаются и заказчиками (бизнесом) и разработчиками.

При написании пользовательской истории следует уделять внимание контексту и всем поведенческим особенностям. Рассказ ведется не от лица абстрактного пользователя или персонажа, а от лица реального человека (либо от имени того, кто ведет наблюдение, проводит интервью). Пример пользовательской истории.

«Выиграли билеты на концерт Radiohead в Риме, сборы проходили в последний момент, нанервничались и чуть не поругались :) Решили, лететь налегке.

Когда поехали в аэропорт, то очень торопились и решили не заезжать на заправку, чтобы сэкономить время. В тот момент нам казалось, что бензина достаточно и хватит добраться до аэропорта, но в наши планы вмешались непредвиденные обстоятельства.

Вобщем,встряли в пробку и, спустя некоторое время, загорелась лампочка низкого уровня топлива. Пришлось спешно искать заправки поблизости. Как назло, нашу машину можно заправлять только 98-ым бензином, что только добавило накала страстей.

По итогу, нам повезло — заправку нашли, в аэропорт успели, но пришлось сильно поволноваться. Благо Том Йорк сгладил впечатления о поездке, да и Рим крут.» [1]

Концептуальный сценарий (Conceptual scenario)

Концептуальный сценарий предполагает уменьшение описания контекста, абстрагирование от незначимых деталей (чей концерт и в какой город собирались лететь), упрощение, более четкое и конкретное описание сути задачи. Все малозначительное отбрасывается, похожие «легенды» объединяются в одну. Концептуальные сценарии важны для генерации идей,выявления ключевых требований и поиска ответа на вопрос «Как улучшить существующий опыт?».Технических подробностей в концептуальном сценарии нет. О завершении этапа создания концептуального сценария можно говорить тогда, когда в процессе обсуждения возникает определенная система, с которой взаимодействует пользователь. Пример концептуального сценария [1].

«Люди в стрессовых ситуациях нуждаются в дополнительном внимании и заботе. В процессе ввода внавигатор конечного адреса маршрута можно определять расстояние до финиша, проверять текущий уровень топлива с учетом актуального среднего расхода и определять, хватит ли нам топлива добраться до заданной точки. Если нет, оповещать об этом водителя.

Если же лампочка низкого уровня топлива уже загорелась, то показывать в навигаторе ближайшие заправки (с возможностью сортировки голосовыми командами, например, по октановому числу)»[1].

Конкретный сценарий (Concrete scenario)

На этапе создания конкретного сценария важна сегментация аудитории, наличие профилей персонажей. На основе этих усредненных характеристик целевой аудитории и прописывается путь достижения цели. То есть конкретный сценарий пишется от третьего лица – от лица определенного персонажа. Конкретный сценарий включает в себя детали реализации проекта и технические подробности. В конкретный сценарий также можно включить ограничения: например, пользователь делает заказ с ПК, мобильной версии сайта или из приложения.

Чтобы проработать весь процесс взаимодействия покупателя с сайтом, нужно создать нескольких персонажей, продумать для каждого из них историю, концепт и конкретный сценарий. После этого конкретные сценарии объединяют по главному признаку. Конечный вариант, который получился после такого объединения, называется сценарием применения или кейсом. Пример конкретного сценария.

«Когда Сергей вводит адрес в навигатор, то он хочет быть уверенным, что ему хватит топлива доехать до указанной точки. Если нет, то автомобиль предупредит об этом Сергея. Если в момент движения Сергей отклонится от маршрута и поедет другим путем (или будет ехать очень быстро), то с включением лампочки низкого уровня топлива он увидит на карте навигатора ближайшие заправки и сможет выбрать подходящую по особенностям топлива. Во время движения соединение с интернетом может прерываться.»[1].

Вариант использования (Use case)

Вариант (пример) использования, кейс применения (Use case) — это пошаговый алгоритм, описывающий взаимодействие пользователей и системы, кто, что, как и в какой последовательности делает. В центре внимания здесь находятся функции, а не чувства и мысли пользователя. Действующим лицом в вариантах использования является не персонаж, а абстрактная усредненная пользовательская роль, например, новый пользователь, администратор магазина, зарегистрированный клиент и т.п.

«Я, как пользователь-водитель (с включенной лампочкой низкого уровня топлива) смогу:

- посмотреть все ближайшие заправки на карте;

- посмотреть все ближайшие заправки списком;
- выбрать заправки нужного бренда;
- посмотреть на выбранной заправке наличие нужного топлива;
- построить до выбранной заправки маршрут».

Часто варианты использования пишут по стандартизированному шаблону.

Таблица 1. Пример варианта использования [2]

Краткое описание функции	Заказ покрывала
Первичные участники	Пользователь (зарегистрированный / незарегистрированный)
Предусловие	Открыта карточка товара
Постусловие	Заказ отправлен онлайн-менеджеру или администратору на офлайн-точку
Основной сценарий	
№ шага	Действие
1	Пользователь открывает страницу «Покрывала»
2	Система открывает каталог покрывал
3	Пользователь выставляет фильтры – цену, материал, цвет, наличие отзывов, сортирует по популярности
4	Пользователь открывает карточку товара
5	Пользователь читает отзывы
6	Пользователь добавляет товар в корзину
7	Пользователь заполняет форму для курьерской доставки
8	Пользователь оплачивает заказ и доставку картой
Альтернативный сценарий – самовывоз	
№ шага, № подшага	Действие
7	Пользователь выбирает точку самовывоза
7a	Пользователь выбирает вид оплаты – картой на сайте или в точке самовывоза

Пользовательские истории Agile методологии (Agile User story)

Такие сценарии пишутся по схеме «Как, <роль пользователя>, я <хочу получить>, <с такой-то целью>». Пример такой истории [1]: «Я, как водитель, в момент включения лампочки низкого уровня топлива, хочу заправить свою машину на ближайшей заправке.» В такой истории

1. есть один персонаж (роль пользователя);
2. есть одно действие;
3. есть одна ценность (value, impact).

Персонажи должны быть созданы до написания User story. Действие – это суть истории, «что нужно сделать». Действие должно быть одно — основное. Не нужно описывать «авторизуется и выполняется поиск» или «указывает параметры поиска и выполняет поиск». В Agile методологии принято указывать только одно действие. Лучше написать несколько маленьких историй, чем одну большую. Важно описывать историю на уровне «ЧТО?» делает, а не «КАК?», т.е. в истории не должно быть упоминания конкретных элементов интерфейса.

Ценность – самая сложная часть при написании User Story. Можно отказаться от формулировки «чтобы» и перейти с понятия ценности (value) на влияние (impact). История не обязательно должна иметь ценность, но обязательно должна оказывать влияние на того персонажа, что указан в истории. А уже это влияние ведет в конечном итоге к цели, которая имеет ценность.

В Agile User Story формулирует не только ценность, но и требования для разработки и тестирования. Для этого к тексту User Story можно добавить критерии приемки, технические заметки, описание обработки ошибок.

Например,

«Как водитель с загоревшейся лампочкой бензина я хочу быстро найти ближайшую хорошую заправку, чтобы заправиться качественным бензином.

Критерии приемки:

1. Как водитель с загоревшейся лампочкой я могу просмотреть все ближайшие заправки.
2. Как ... я могу выбрать заправки подходящих мне брендов АЗС.
3. Как ... я могу видеть ближайшие заправки выбранных брендов списком.
4. Как ... я могу видеть ближайшие заправки выбранных на карте.

Обработка ошибок:

1. При выключенной геолокации пользователя необходимо дать ему информацию о том, где ее включить.

Технические заметки:

1. Заправки в списке должны обновляться при изменении местоположения пользователя на 100 метров.» [3].

Такая формулировка помогает разработчикам и тестировщикам смотреть на основные проблемы, которые должны быть решены в рамках User Story.

Все виды сценариев создаются для того, чтобы выявить требования к проектируемой системе: что система должна делать, чтобы пользователь мог достичь цели, решить задачу. Чтобы проработать в пользовательском сценарии все важные для пользователя моменты, необходимо пройти все 4 этапа, от создания истории до кейса. Пользовательские истории и концептуальные сценарии нужны для понимания основных мотивов пользователей и погружения в их карту реальности. Конкретные сценарии и варианты применения могут использоваться для проектирования информационной архитектуры, интерфейсов, при проведении тестирования.

Написание сценариев требует дополнительных усилий и времени, поэтому при проектировании простых информационных систем можно отказаться от сценариев, а при проектировании систем, где взаимодействие достаточно простое и прямолинейное, отсутствуют альтернативные и исключительные сценарии (или их проще и быстрее отобразить в прототипе) можно обойтись без вариантов использования. Но чем более сложным является взаимодействие «пользователь-интерфейс», тем больше необходимость в написании всех 4 видов сценариев. Иначе есть риск вложить 80% усилий в 20% сценариев, с которыми пользователи будут работать мало и редко, что снизит качество продукта.

На следующем этапе проектирования все истории, сценарии и варианты использования объединяют и создают User Flow Map.

Задания на лабораторную работу

1. Изучите теоретические сведения
2. Напишите 4 вида пользовательских сценариев для проектируемой системы.
3. Оформите отчет.
4. Защитите лабораторную работу.

Содержание отчета

1. Титульный лист.
2. Цель работы.
3. Краткие теоретические сведения.

4. Пользовательские истории, концептуальный сценарий, конкретные сценарии, варианты использования проектируемого приложения.
5. Выводы по работе.

Контрольные вопросы

1. Что представляет собой пользовательский сценарий (UserStory)?
2. Зачем нужны пользовательские сценарии?
3. Какие виды пользовательских сценариев Вы знаете?
4. Охарактеризуйте различные виды пользовательских сценариев.
5. В чем заключается специфика пользовательских сценариев, создаваемых по Agile методологии?
6. Как определить, какой вид пользовательского сценария нужно использовать?

1. Гуров, А. «Userstories vs Usecases vs Scenarios vs Agileuserstories vs?»[Электронный ресурс]. / А.Гуров. – Режим доступа: <https://medium.com/design-pubразница-между-user-stories-и-use-cases-5f52381150fa>.

2. Демахина, Т. Пользовательские сценарии: что это, как и для чего их нужно строить. Как сделать сайт удобным для пользователей.[Электронный ресурс]. / Т. Демахина –

Режим доступа: <https://l-a-b-a.com/blog/2651-polzovatelskie-scenarii-cto-eto-kak-i-dlya-chego-ih-nuzhno-stroit>.

3. Минникова, А. Как правильно написать User Story? [Электронный ресурс]. / А. Минникова. – Режим доступа: <https://apptractor.ru/develop/user-story-plan-deystviy-dlya-razrabotchika.html>.