

# Человеко-машинное взаимодействие

## Лекция 4

### UX

#### Опыт взаимодействия

**Опыт пользователя, опыт взаимодействия (User eXperience, UX)** – это восприятие и ответные действия пользователя, возникающие в результате использования и/или предстоящего использования продукции, системы или услуги.

**User Expirience** – все аспекты взаимодействия конечного пользователя с компанией, её услугами и продукцией. – Д. Норман

**UI - User Interface, интерфейс пользователя** – интерфейс, обеспечивающий передачу информации между пользователем-человеком и программно-аппаратными компонентами компьютерной системы.

UX более субъективное понятие, чем интерфейс пользователя потому что включает в себя ещё и реакцию пользователя.

#### UX vs UI

Программа может иметь хороший, удобный и понятный UI, но долго запускаться, неожиданно закрываться, иметь не весь важный для своей области применения функционал, а значит UX может быть не таким же хорошим.

С другой стороны программа может соответствовать ожиданиям пользователя, вызывать хорошие эмоции при не слишком выверенном UI .

Таким образом можно создать положительную субъективную оценку от программного продукта, который имеет относительно серьёзные недостатки.

#### UX и когнитивные искажения

Нередко хороший или плохой UX становится следствием когнитивных искажений.

Эффект IKEA – это когнитивное искажение, которое появляется, когда покупатели непропорционально высоко оценивают значимость (ценность) товаров, которые они создают отчасти сами (например, собирают из деталей).

#### Пример улучшений

Выполнение длительной операций (запуска программы, скачивание обновления, сохранение настроек) воспринимается пользователем приятнее, если он видит индикацию прогресса. Например полосу прогресса, изменяющиеся сообщения, и т.п.

Например: запуск slack, загрузка youtube, доставка почты в почтовых программах.

Некоторые операции, которые выполняются по мнению пользователей слишком быстро также искусственно замедляют. Из-за мгновенного выполнения важной операции, у пользователя может сложиться ощущение, что она не выполнена или выполнена с ошибкой.

## **Как проектировать UX?**

Пять уровней UX – это концептуальная модель, предложенная Джессом Гарреттом (Jesse James Garrett ) для **проектирования опыта пользователя** веб-приложений.

Процесс разработки и планирования UX начинается с наиболее абстрактного (общего) уровня - **стратегии**, и постепенно проходит через все уровни до самого конкретного – **поверхности**.

### **Уровни UX**

- Уровень поверхности (surface)
- Уровень компоновки (skeleton)
- Уровень структуры (structure)
- Уровень возможностей (scope)
- Уровень стратегии (strategy)

#### **Уровень стратегии**

Наиболее общее описание продукта включающее:

- Цели пользователя текущие и возможные
- Цели заказчика
  - Назначение ПО
  - Соответствие ПО бренду

На этом этапе нужно ответить на вопрос: Для чего разрабатывается продукт?

### **Метод персонажей**

- Нужно знать своих потенциальных пользователей, их цели и нужды
- Часто целевая аудитория программы неоднородна, поэтому усреднённый пользователь будет слишком нереалистичной моделью
- Целевую аудиторию разделяют на несколько групп по принципу схожести целей и опыта пользователей
- Для каждой группы описывается типичный представитель – персона (персонаж)
- Персона описывается так, если бы это был живой человек. Это способствует эмпатии UX дизайнера по отношению к пользователю
- Метод персон используется для того чтобы понять, что может заинтересовать, а что, наоборот, оттолкнёт потенциального пользователя.
- Используется в техзадании, чтобы разработчики могли учесть интересы разных категорий людей.
- Персонажи строятся на основе исследования пользователей.

#### **На основе чего создавать персонажа?**

- Информация о целевой аудитории может быть у заказчика
- Изучите самих пользователей

## **Персонаж**

- Описание персонажа (имя, пол, возраст, работа... )
- Потребности
- Демотивирующие факторы – что отпугивает или расстраивает пользователя
- Опыт
- Контекст использования, сценарии
- ...

## **Метод персонажей**

- Персонаж используется на всём цикле разработки ПО.
- На основе потребностей персонажей создаётся диаграмма вариантов использования (диаграмма прецедентов).
- На основе сценариев разрабатываются пользовательские сценарии. С их помощью оценивается информационный дизайн и компоновка.
- Для каждого сценария персонажа указывается частота его выполнения. Оптимизируются соответствующие пользовательские сценарии и UI.
- Уровни структуры, компоновки и поверхности разрабатываются с учётом ментальных моделей пользователя, его опыта и демотивирующих факторов.

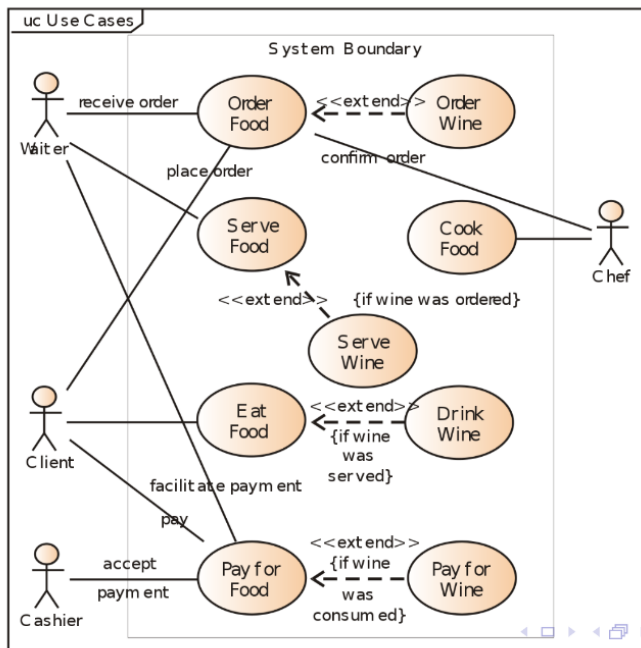
## **Уровень возможностей**

Составляется список функциональных возможностей, контент сайта.

На этом этапе нужно ответить на вопрос: Что разрабатывается?

## **Диаграмма прецедентов**

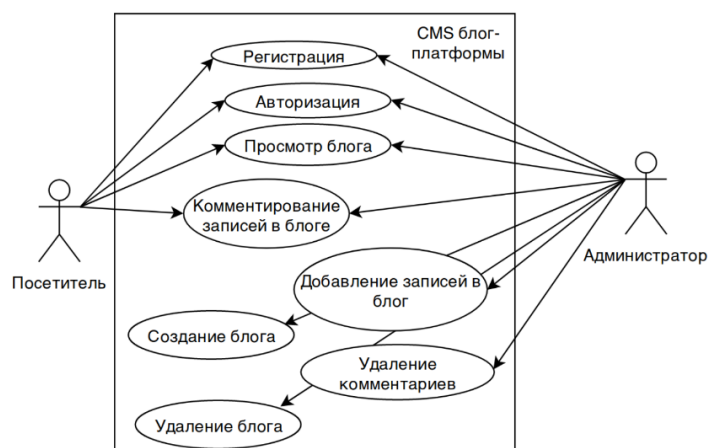
Для описания функциональных возможностей может использоваться use case диаграмма (диаграмма прецедентов).



- Диаграмма строится из 3 основных элементов:
- Система (рамки системы). Прямоугольник обозначающий систему. Включает в себе прецеденты. Иногда не приводится
- Прецедент – возможность системы (часть её функциональности), благодаря которой пользователь может получить конкретный, измеримый и нужный ему результат. Обозначается овалом с описанием возможности из 1-3 слов. Например: просмотр страниц сайта, комментирование, добавление постов
- Актёр (actor) – роль, которую пользователь играет по отношению к системе. Обозначается человечком с названием роли Например: посетитель сайта, администратор сайта

Таким образом диаграмма вариантов использования (use case diagram, диаграмма прецедентов) – UML-диаграмма, отражающая отношения между актёрами и прецедентами.

#### Диаграмма прецедентов. Пример 1



Пример диаграммы прецедентов для простой CMS

