**Сбор и анализ требований**

Основные вопросы, решаемые на этом этапе:

1. *Цели и задачи бизнеса*

1.1 Какие проблемы решает система?

1.2 Что она улучшит и автоматизирует?

2. *Заинтересованность в По пользователя*

2.1 Кто основные пользователи?

2.2 Какие роли пользователя

2.3 Функциональные возможности каждой роли

3. *Функциональные требования*

3.1 Какие действия выполняются в системе

3.2 Должна ли система интегрироваться с существующей

4. *Нефункциональные требования*

4.1 Смотри качественные требования к ПО

5. *Требования к данным*

5.1 какими типами данными и форматами данных должна работать система

5.2 какими должны быть входные и выходные данные

5.3 у каких пользователей какие данные и что они могут использовать,

5.4 какие данные необходимо хранить и обрабатывать

5.5 требуется ли выполнять архивирование данных

*6. Архитектура*

6.1 Какой тип информационной системы требуется разработать

6.2 Какой предпочтительный технологический стэк и платформа этой системы

6.3 Имеется ли возможность обновления программных и аппаратных средств

*Способы сбора требований:*

1) Интервью

2) Анкетирование или опрос

3) Совещание

4) Мозговой штурм

5) Автозапись – изучение документов, заметок и записок, аудиозаписей, присланных заказчиком

6) Изучение существующей документации, используемой в организации заказчика

7) Повторное использование существующей документации (спецификации)

8) Представитель заказчика в команде разработчика

9) Представитель команды разработки в команде заказчиков

10) Обучение

*Сценарий вариантов использования:*

1) use-case

2) user-story

● Все требования обязательно фиксировать или записывать

● Оформление результатов выполняется в виде ТЗ

Виды документов:

1) ТЗ – техническое задание

2) Эскизный проект

3) Технический проект

● Оформление и согласование документов обязательно при участии заказчика

ТЗ – это документ в котором выделяются и фиксируются результаты требований, анализы предметной области и информационной системы.

● Результаты сбора и анализа требований можно представить в виде диаграммы

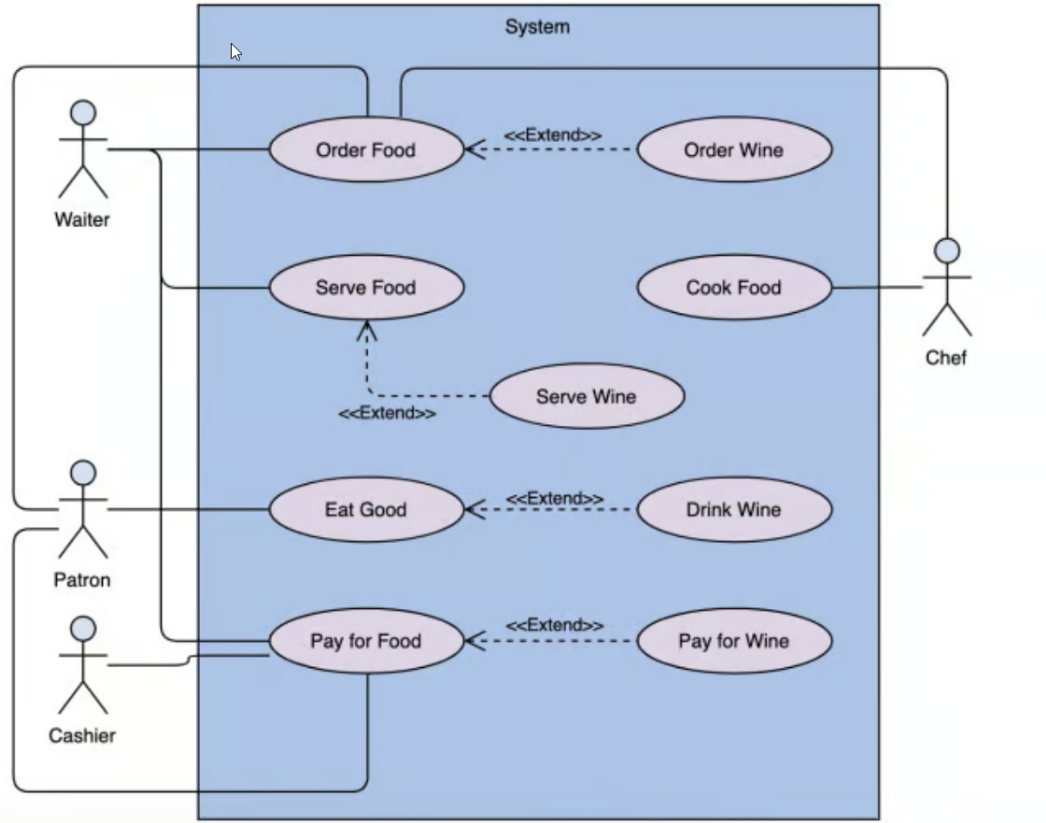
Диаграмма варианта использования (Use-case диаграмма) – это UML диаграмма, показывающая отношения между акторами или вариантами использования. Основная задача диаграммы: показать возможности каждой категории пользователей в системе (заказчику или разработчику).

*Элементы диаграммы:*

1) actor – отображается как человечек, означает набор ролей пользователей. (Может быть человек, внешняя сущность, класс, другая система и т.д.) с друг другом связаны только при отношении наследования или обобщения

2) Вариант использования, прецедент, Use-case. Обозначается эллипсом внутри которого действие, выполняемое системой. Результат действия отображается actor.

● надпись описывает то, что делает система. Указывается глаголом или отглагольным существительным

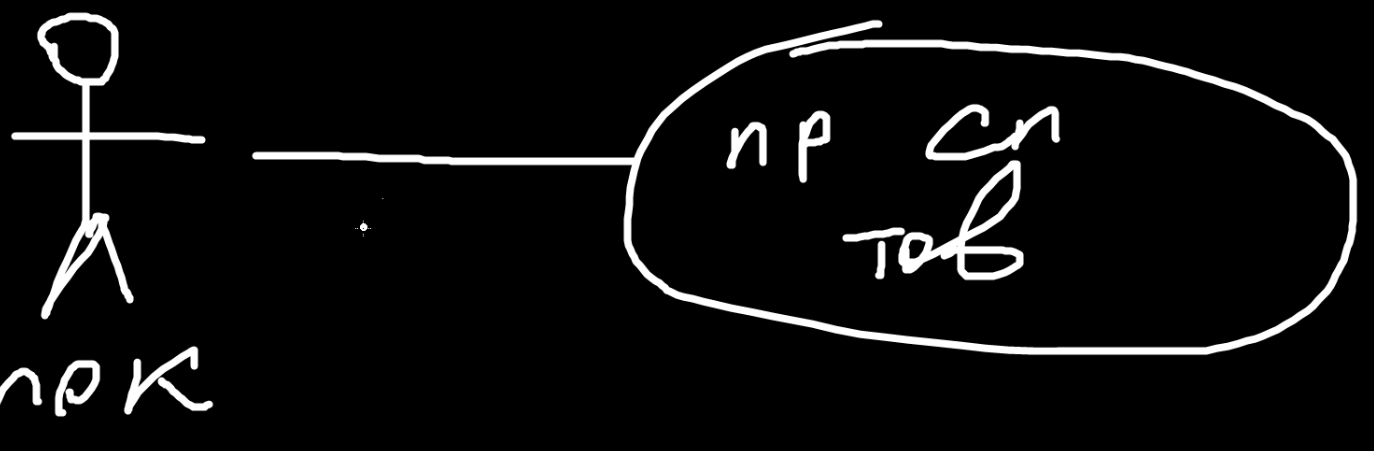


Вариант использования может быть связан с другими вариантами.

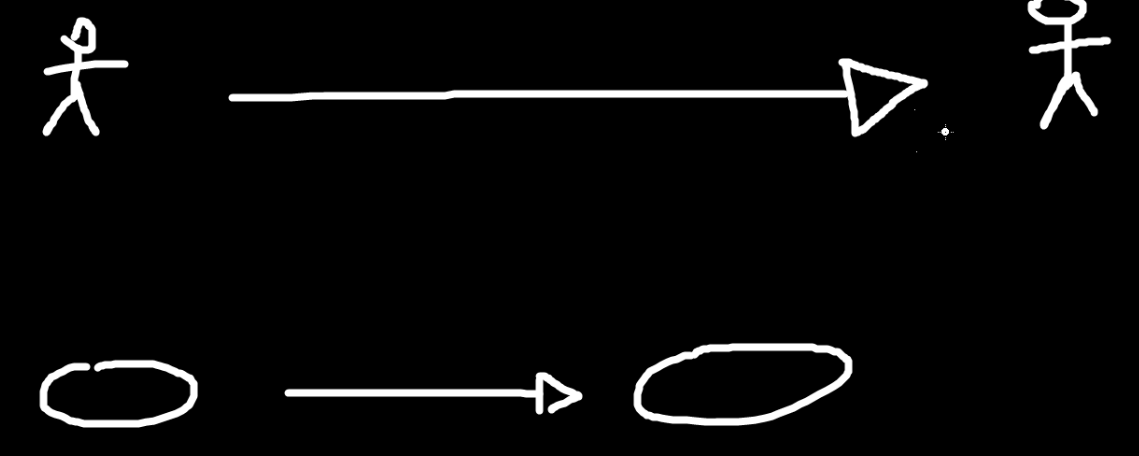
3) Рамки системы – прямоугольник с названием в верхней части, и прецедентами внутри. Могут не использоваться.

*Виды отношений или связей:*

*1*) Ассоциация – взаимодействия актора или прецедента



2) Обобщение или наследование. Один элемент диаграммы является частным случаем другого.



Стрелка показывает на родителя, т.е. на общий случай.

Include – вариант использования содержит поведение, определенного в другом варианте использования. Включаемый вариант должен выполнится, как часть основного варианта.



Extend – один вариант использования связан с другим, расширяющий основной вариант дополнительной функциональностью . Стрелка показывает на основной вариант.