**Сбор и анализ требований**

**Основные вопросы, решаемые на этом этапе:**

1. Цели и задачи бизнеса, какие проблемы должна решить система, что она улучшит и автоматизирует
2. Заинтересованные стороны и пользователи, кто основные пользователи, какие роли пользователей и какие функциональные возможности каждой роли
3. Функциональные требования, какие действия должны выполняться в системе, должна ли интегрироваться система с существующей
4. Нефункциональные требования, смотри критерии качества ПО\*
5. Требования к данным, с какими типами данных и форматов файлов должна работать система, какими должны быть входные и выходные данные, у каких ролей пользователя к каким данным должен быть доступ, какие данные необходимо хранить и обрабатывать, требуется ли выполнять архивирование данных
6. Архитектура ИС, какой тип ИС требуется разработать
7. Технические ограничения, каков предпочтительный технологический стек и платформа для этой системы, имеется ли возможность обновления параметров аппаратных средств

**Способы сбора требований:**

1. Интервью
2. Анкетирование или опрос, можно предлагать выбор или дать возможность написать свой ответ
3. Совещание
4. Мозговой штурм
5. Автозапись - это изучение документов, заметок и записок, аудиозаписи присланных заказчиком
6. Изучение существующей документации используемой в организации заказчика
7. Повторное использование спецификации
8. Представитель заказчика команде разработчика
9. Работа в “поле” — это представитель команды разработки в команде заказчика
10. Обучение – представитель команды разработчиков обучается бизнес процессам у заказчика

**Сценарии вариантов использования:**

* User story – я, как категория пользователей, хочу иметь возможность, что бы получить результат или ценность
* Use case

Все требования обязательно фиксировать

Оформление результатов обычно выполняется в виде ТЗ, то есть по требованиям гост на тех. Документацию

**Виды документов:**

1. Техническое задание
2. Эскизный проект
3. Технический проект

Оформление и согласование документов обязательно при участии заказчика

**ТЗ** ­­- это документ в котором определяются и фиксируются результаты сбора требований, анализа предметной области и ИС

Результаты сбора анализа требований, можно представить в виде диаграммы

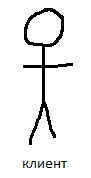
**Диаграмма вариантов использования (use case diagram)**

Это UML диаграмма, показывающая отношение между акторами или вариантами использованиями.

**Основная задача диаграммы** – показать возможности каждой категории пользователей в системе, заказчику или разработчику

**ПО для создания use case diagram**:

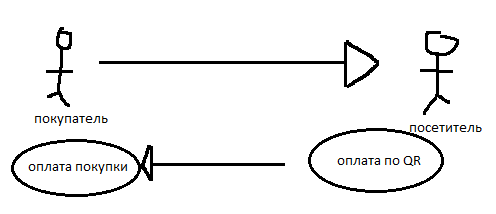
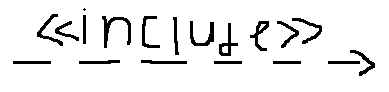
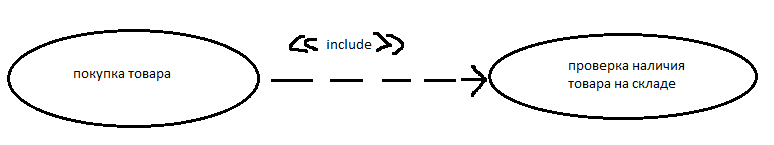
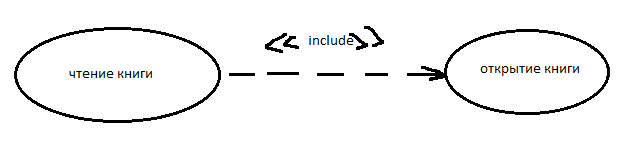
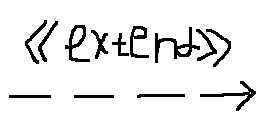
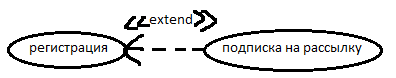
* MS Visio
* Draw.io

**Элементы диаграммы:**

1. Actor, отображается как человечек, обозначает набор ролей пользователей, может быть человек, внешняя сущность, класс, другая система. Актеры могут быть связаны друг с другом только при отношении наследовании или обобщения
2. Вариант использования/прецедент/use case – обозначается эллипсом, внутри которого доступное действие, выполняемое системой, результат действия отображается актором, надпись описывает то, что описывает система, указывается глаголом или отглагольным существительным, или словосочетанием с ними. Варианты использования могут быть связаны с другими вариантами.
3. Рамки системы – это прямоугольник с названием в верхней части и прецедентами внутри могут не использоваться



**Виды отношений или связи:**

1. Ассоциации – это взаимодействия актора и прецедента
2. Обобщение или наследование – предназначена для уточнения, что один элемент диаграммы, частный случай другого, стрелка показывает на родителя, то есть на общий случай
3. Включение/include – вариант использования содержит поведение определённого в другом варианте использования, включаемый вариант должен выполнится как часть основного варианта, включение позволяет избежать дублирование кода и повторно использовать функциональность, стрелка показывает на включаемый вариант без которого не выполнить основной
4. Отношение расширения/extend – один вариант использования связан с другим, расширяющим основной вариант доп. Функциональностью, расширение позволяет добавлять новую функциональность не меняя основной вариант, стрелка показывает на основной вариант который может быть расширен

