

ПРОГРАММНАЯ ИНЖЕНЕРИЯ

ОПРЕДЕЛЕНИЕ ТРЕБОВАНИЙ.

ТРЕБОВАНИЯ

ОПРЕДЕЛЕНИЕ ТРЕБОВАНИЙ – ОСНОВА ПРОЕКТИРОВАНИЯ СИСТЕМЫ

- ◎ Большая часть работы с требованиями проводится в начале проекта, в фазах Начало и Уточнение.
- ◎ До начала работы над ОО анализом и проектированием уже необходимо иметь представление о том, что должно получиться в результате.
- ◎ Выработка требований – это переговорный процесс, поскольку часто выдвигаются противоречивые требования, которые должны быть согласованы.

ТРЕБОВАНИЯ И ДИАГРАММА ПРЕЦЕДЕНТОВ

- ◎ Некоторые курсы UML утверждают, что прецеденты UML являются единственным способом фиксирования требований.
 - ◎ Прецеденты могут отражать только функциональные требования, которые являются описанием того, *что будет делать система.*
- ◎ Но существуют требования которые не относятся к функциональности, но являются ограничениями, накладываемыми на систему (производительность, надежность)

ТРЕБОВАНИЕ

- ◎ Требование – это подробное описание того, что должно быть реализовано.
- ◎ Функциональные требования – какое поведение должна предоставлять система;
- ◎ Нефункциональные требования – особое свойство или ограничение, накладываемое на систему

Требования указывают что должно быть построено, но не говорят как это сделать.

ТРАДИЦИОННОЕ ОПИСАНИЕ ТРЕБОВАНИЙ



ПРИМЕРЫ ФУНКЦИОНАЛЬНЫХ ТРЕБОВАНИЙ

- ① 1. Система АТМ **должна** проверять действительность вставленной в банкомат карточки.
- ① 2. Система АТМ **должна** проверять достоверность PIN-кода, введенного пользователем.
- ① 3. Система АТМ **должна** выдавать по одной АТМ-карточке не более \$250 в сутки.

ПРИМЕРЫ НЕФУНКЦИОНАЛЬНЫХ ТРЕБОВАНИЙ

- ① 1. Система АТМ **должна** быть написана на С++.
- ① 2. Система АТМ **должна** обмениваться информацией с банком, используя 256-разрядную кодировку.
- ① 3. Система АТМ **должна** проверять действительность карточки АТМ в течение не более трех секунд.
- ① 4. Система АТМ **должна** проверять достоверность PIN-кода в течение не более трех секунд.

АТТРИБУТЫ ТРЕБОВАНИЙ

Набор критериев важности «MoSCoW»

Значения атрибута Priority	Семантика
Must have (обязан иметь)	Обязательные требования, являющиеся фундаментальными для системы.
Should have (должен иметь)	Важные требования, которые могут быть опущены.
Could have (мог бы иметь)	По-настоящему необязательные требования (реализуются, если есть на это время).
Want to have (хотел бы иметь)	Требования, которые могут подождать до следующих версий системы.

ПОИСК ТРЕБОВАНИЙ

ПОИСК ТРЕБОВАНИЙ

- ◎ непосредственные пользователи системы
- ◎ другие заинтересованные стороны (например, руководители, специалисты обслуживания, установщики)
- ◎ другие системы, с которыми взаимодействует данная система
- ◎ аппаратные устройства, с которыми взаимодействует данная система
- ◎ правовые и регулирующие ограничения
- ◎ технические ограничения
- ◎ коммерческие цели

ВЫЯВЛЕНИЕ ТРЕБОВАНИЙ

Естественный язык формируют 3 фильтра:

- ⊙ Пропуск – информация отфильтровывается
- ⊙ Искажение – информация изменяется взаимосвязанными механизмами вымысла и представления
- ⊙ Обобщение – информация обобщается в правила, убеждения и понятия об истинности и ложности

ПРИМЕР «ПРОПУСКА»

- ◎ **Пример:** «Они используют систему для получения книг на время» – *пропуск*.
- ◎ **Вопрос:** Кто именно использует систему для получения книг на время?
- ◎ **Ответ:** читатели, другие библиотеки и библиотекари.

ПРИМЕР «ИСКАЖЕНИЯ»

- ◎ **Пример:** «Тот, кто имеет книгу на руках, не может взять другую книгу до тех пор, пока не вернет предыдущую, срок возврата которой истек» – *искажение*.
- ◎ **Вопрос:** Существуют ли такие обстоятельства, при которых кто-либо мог бы взять новую книгу до того, как будут возвращены все имеющиеся на руках книги, срок возврата которых истек?
- ◎ **Ответ:** Фактически существует два обстоятельства, при которых право читателя на получение книг может быть восстановлено. Во первых, все имеющиеся на руках книги, срок возврата которых истек, возвращены; во-вторых, за все невозвращенные книги, срок возврата которых истек, внесена плата.

ПРИМЕР «ОБОБЩЕНИЯ»

- ◎ **Пример:** «Для получения книг у всех должен быть формуляр» – *обобщение*.
- ◎ **Вопрос:** Есть ли пользователи системы, которым не обязательно иметь формуляр?
- ◎ **Ответ:** Некоторые пользователи системы, например другие библиотеки, могут не иметь формуляра или имеют специальный формуляр с другими сроками и условиями возврата книг.

КВАНТОР ОБЩНОСТИ

◎ При встрече с квантором общности всегда можно найти пропуск, обобщение и искажение

- **все**

- **каждый**

- **всегда**

- **никогда**

- **никто**

- **несколько**

ИНТЕРВЬЮ

- ⊙ Не заблуждайтесь, вам может *казаться* что вы все очень хорошо понимаете;
- ⊙ Задавайте контекстно-свободные вопросы;
- ⊙ Не занимайтесь телепатией. Телепатия – это **заблуждение** по поводу того, что вам известны чьи-то чувства.
- ⊙ Лучший способ записи информации во время интервью – бумага и ручка!
- ⊙ Анкеты не заменяют интервью, но могут быть полезным дополнением.

- ◎ **UML (Unified Modeling Language)** - это универсальный язык визуального моделирования систем
- ◎ **Унифицированный процесс разработки программного обеспечения (Unified Software Development Process, USDP)** – это SEP от авторов UML.
- ◎ В UP четыре фазы: Начало, Уточнение, Построение, Внедрение. Каждая фаза может состоять из 1 или более итераций. Каждая итерация состоит из определения требований, анализа, проектирования, реализации, тестирования.

- ◎ Определение требований – основа проектирования системы.
- ◎ Требование – это подробное описание того, что должно быть реализовано
- ◎ Бывают Функциональные и нефункциональные требования
- ◎ Для выявления требований используют интервью. При проведении интервью необходимо учитывать 3 **фильтра**: пропуск; искажение; обобщение.