

## **Лабораторная работа №7-8**

### **Определение границ системы**

**Цель:** определить границы разрабатываемой системы «Университет»: что программа будет делать, и что делать не будет.

#### **Теоретические вопросы**

Все требования в проекте делятся на две основных категории: потребности (пользователей) и решения (функции, feature), которые будут реализованы в проекте для удовлетворения потребностей.

Если в процессе работы над продуктом требуется изменить решение, то достаточно утверждения нового решения проектной командой. Если изменениям подвергается потребность, то это обязательно должно быть обговорено с клиентом.

Требование определяется как «условие или особенность, которой должна удовлетворять система» [5].

Требованием может быть:

- функциональность, необходимая заказчику или пользователю для разрешения проблем (получения прибыли);
- функциональность, которая должна быть реализована в системе в соответствии с контрактом, стандартом, спецификацией, инструкцией или другим официальным документом;
- ограничение, наложенное заинтересованным лицом.

Типы требований, наиболее часто использующихся в проектах:

- потребности заинтересованного лица: требование от заинтересованного лица;
- функциональная особенность: функциональность, предоставляемая системой; формулируется системным аналитиком; назначение – удовлетворить потребности заказчика;
- сценарий использования (Use Case): описание поведения системы в терминах последовательности действий;
- дополнительное требование: другое требование (обычно нефункциональное), не охваченное сценариями использования;
- тестовые сценарии (Test Cases): спецификация тестовых исходных данных, условий выполнения и ожидаемых результатов;

– сценарий (Алгоритм, Scenario): особая последовательность действий; определенный путь по сценариям использования.

Их взаимосвязи отражены с помощью пирамиды требований (рис. 14).

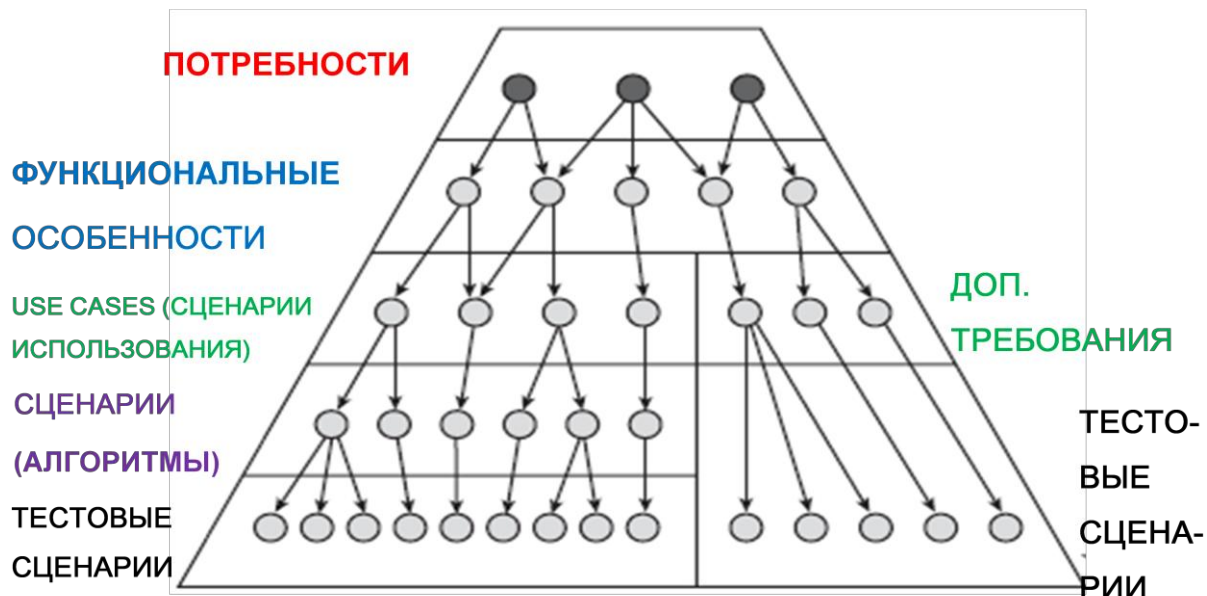


Рис. 14. Пирамида требований

Главное отличие между потребностями и функциональными особенностями заключается в источнике требований. Потребности озвучиваются заинтересованными лицами, а функциональные особенности формулируются аналитиками.

Последовательность этапов от выявления потребностей до разработки тестовых сценариев, т.е. сверху вниз, является процессом проектирования системы. Движение по пирамиде от тестовых сценариев и удовлетворения потребностей пользователя, т.е. снизу вверх, демонстрирует работу системы.

Рассмотрим процесс формирования сценариев использования с помощью диаграммы прецедентов языка UML.

В нотации языка UML данный вид диаграмм называется по-разному: диаграммы сценариев использования, вариантов использования или диаграммы прецедентов (английский вариант – Use Cases). Это диаграмма взаимодействия пользователя (актера, эктера, актёра, англ. actor) с системой для получения результата.

Здесь эктор – находящееся вне системы нечто (некто), взаимодействующее с системой. Это может быть как пользователь, так и другая программа. При взаимодействии пользователя с системой первый ожидает определенного, предсказуемого действия последней. В свою очередь

прецедент (сценарий использования) выступает одним из способов определения реакции системы [2]. С помощью прецедентов описывают поведение разрабатываемой программы, не определяя ее реализацию.

С помощью наглядного изображения диаграмм можно достичь взаимопонимания между заинтересованными лицами проекта: разработчиками, аналитиками, пользователями и др., – а также проверить архитектуру системы во время ее разработки.

Составление диаграммы прецедентов – это способ перейти от функциональных особенностей к конкретным сценариям. Каждый отдельный прецедент представляет функциональные требования в целом и отражает выполнение некоторого объема работ. Для конечного пользователя – эктора – прецедент реализует нечто ценное, например, формирует отчет, вычисляет, обрабатывает данные и проч. Прецеденты в свою очередь не должны быть слишком общими или чересчур специфичными.

Изображение системы на диаграмме осуществляется в виде прямоугольника. Прецедент – в виде эллипса, находящегося в границах системы. Для каждого прецедента должно быть определено уникальное имя – текстовая строка внутри эллипса. Эктор – на рис. 15.



Рис. 15. Нотация эктора на диаграмме

Между экторами и прецедентами определяются отношения – ассоциации в виде соединяющих их линий.

**Задание:** нарисуйте диаграмму прецедентов для системы «Оплата сотовой связи».

**Задание:** нарисуйте диаграмму прецедентов для системы «Электронная почта».

Прецеденты делятся на два вида: высокого уровня и развернутые.

Прецеденты высокого уровня (high-level use case) – это очень краткие описания процессов, обычно состоящие из двух-трех предложений. Их используют на начальном этапе при формулировании требований к системе. Это способ осознать степень сложности разрабатываемой системы, которые имеет слабое отношение к конкретному проектному решению.

Развернутые прецеденты (expanded use case) представляют собой более подробное описание, чем прецеденты первого вида. Они позволяют углубить понимание требований и процессов. Обычно оформляются в виде диалога между пользователем и программой.

При этом всегда важно помнить: любой прецедент описывает, что делает система, но не определяет, каким образом она это делает.

### **Ход работы**

**Задание 1.** На основании выбранной вами роли в АИС «Университет» и ранее проведенной ранее работы по выявлению характеристик данного пользователя (должностная инструкция, таблица характеристик (таблица 9), анкета, интервьюирование, анализ аналогов АИС и др.), составьте список требований этого пользователя к программе. Оформите их в виде предложений «система должна». При необходимости возможно повторить интервью для выявления требований пользователя.

**Задание 2.** Работа в группе: составьте полный список требований к системе, удаляя дублирующиеся требования. Из полученного списка выделите требования, которые не подлежат реализации.

**Задание 3.** Для выбранного вами пользователя на основании уточненного в задании 2 списка требований создайте диаграмму прецедентов высокого уровня. Для создания диаграмм можно воспользоваться:

1. Microsoft Visio.
2. Microsoft Visual Studio.
3. Онлайн сервис: <http://creately.com/Draw-UML-and-Class-Diagrams-Online>.
4. Онлайн сервис: <https://www.draw.io/>
5. Онлайн сервис: <http://yuml.me/>.

Если вы работаете в Microsoft Visio, то изучите следующую информацию.

### **Назначение Visio**

Visio представляет собой векторный редактор, являющийся средством построения схем и диаграмм различного назначения. Visio позволяет создавать различные диаграммы, например, IDEF0, DFD, ERD, блок-схемы, диаграммы UML и др.

## Общие сведения об интерфейсе Visio

Вид окна Visio 2013 представлен на рис. 16.

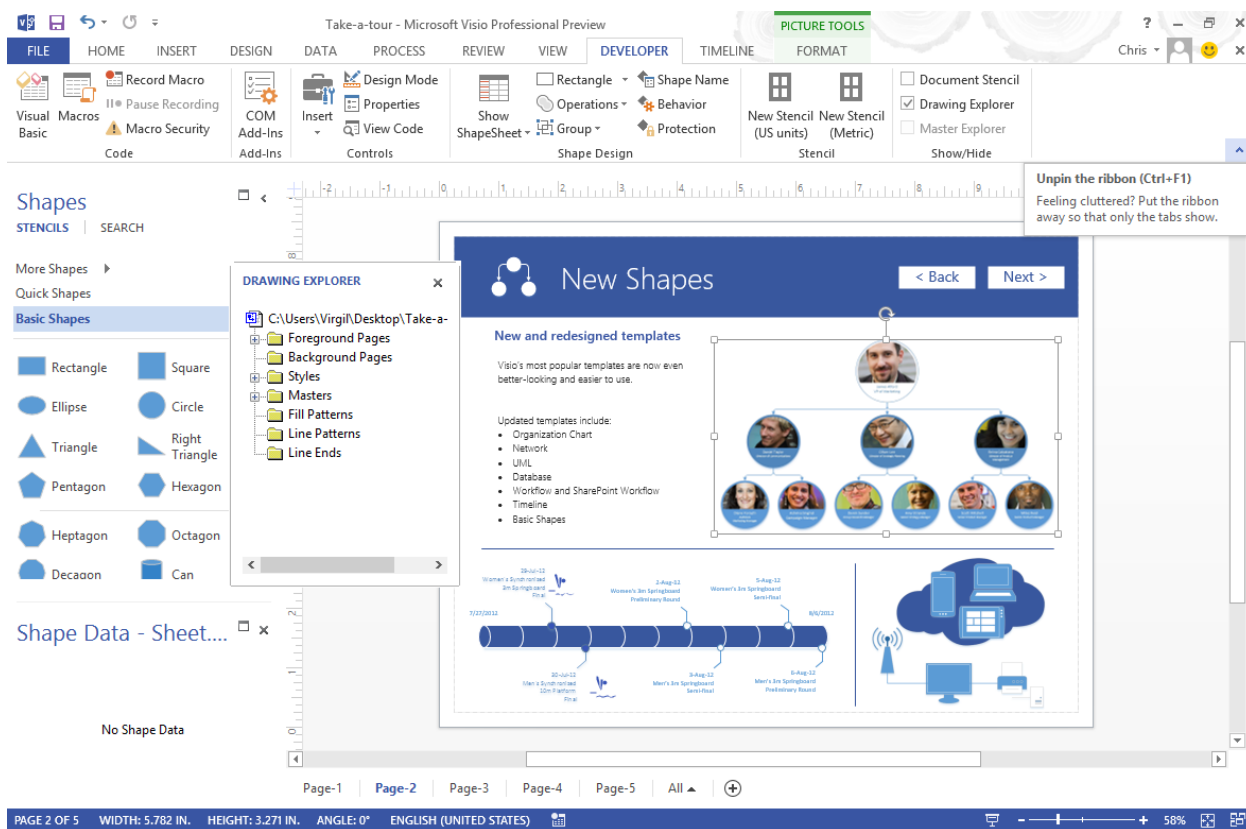


Рис. 16. Векторный редактор Visio

### Создание и редактирование диаграмм UML

1. Запустите программу Microsoft Visio 2013.
  2. При первом запуске выберите опции и нажмите «Принять».
- Откроется окно выбора схемы.
3. Выберите вверху категорию «Программное обеспечение».
  4. Нажмите левой кнопкой мыши по «Схема вариантов использования UML».
  5. Затем «Создать».
  6. С помощью инструментов на панели «Фигуры» создайте диаграмму, перемещая их на рабочее поле методом Drag&Drop.
  7. Сохраните диаграмму.

Рассмотрим некоторые особенности при работе с Visio:

- изменение размеров объекта выполняется с помощью изменения положения прямоугольников, расположенных по контуру объекта. Если

менять положение прямоугольников, расположенных по серединам граней объекта, при нажатой клавише Shift, размеры фигуры меняются пропорционально;

- поворот объекта вокруг его центра выполняется с помощью изменения положения круглой стрелки над объектом;
- для изменения положения, размера или угла наклона группы объектов необходимо вначале выделить эти объекты при нажатой клавише Shift, а затем выполнить требуемое действие;
- для создания надписи внутри объекта выполнить двойной щелчок левой кнопкой мыши при нахождении указателя мыши над объектом.

**Задание 4.** Распределите сформулированные прецеденты между членами рабочей группы. Осуществите документирование прецедентов любым из возможных способов. Например, в виде таблицы (таблица 11).

Таблица 11 – Описание прецедента

Прецедент	Название прецедента (русское и английское).
Исполнители	Исполнители, работающие с прецедентом.
Тип	Какой тип (типы прецедентов будут рассмотрены ниже).
Описание	Словесное описание прецедента, состоящее из двух - трех предложений.

**Задание 5.** Ответьте на следующие вопросы о проекте АИС «Университет»:

1. Кто является поставщиком информации в систему?
2. Кто будет пользоваться информацией из системы?
3. Кто будет удалять информацию из системы?
4. Кто будет управлять системой?
5. Где будет использоваться система?
6. Откуда пользователи будут получать информацию?
7. Имеются ли внешние системы, с которыми программа будет взаимодействовать? Укажите их.

**Задание 6.** После ответов на вопросы из предыдущего задания посмотрите, все ли требования к системе вы учли? При необходимости дополните их и документируйте в виде диаграммы прецедентов.

### Контрольные вопросы

1. Что такое границы разрабатываемой системы?

2. Перечислите методы выявления требования заказчика, пользователей системы.
3. Укажите особенности диаграммы прецедентов UML.
4. Назовите возможные средства создания диаграмм UML. Каковы их преимущества, недостатки?
5. Какого уровня бывают прецеденты?
6. Как осуществляется документация прецедентов?

**По завершении занятия студент должен:**

1. Знать способы выявления требований заказчика, пользователей.
2. Иметь представление о границах разрабатываемой системы.
3. Создавать списки требований к системе.
4. Иметь представление о диаграммах прецедентов UML, их назначении, возможностях при проектировании ИТ-проектов.
5. Создавать диаграммы прецедентов.
6. Добавлять новые варианты отчетов, в том числе с диаграммами.
7. Осуществлять документирование прецедентов высокого уровня.