

## **Лабораторная работа №1**

### **Процессы жизненного цикла ИС. Анализ требований к системе. Процесс заказа.**

#### **1 Цель работы**

Изучить процессы жизненного цикла (ЖЦ) программной системы согласно стандарту ГОСТ Р ИСО/МЭК 12207-99. Получить практические навыки использования стандарта (процесс заказа) во время сбора сведений о требуемой функциональности системы.

#### **2 Порядок выполнения работы**

1. Изучить стандарт ГОСТ Р ИСО/МЭК 12207-99
2. Получить задание на выполнение лабораторной работы согласно своему варианту.
3. Построить диаграммы процесса заказа в нотации IDEF0.
4. Провести анализ требований к системе.
5. Разработать шаблон описания требований на основе историй пользователей.
6. Составить и защитить отчет о лабораторной работе у преподавателя.

#### **3 Содержание отчета**

- титульный лист;
- наименование и цель работы;
- задание на лабораторную работу согласно варианту;
- результаты работы выполнения задания.

#### **4 Краткая теория**

##### **4.1 ЖЦ ИС**

Под *жизненным циклом* системы обычно понимается непрерывный процесс, который начинается с момента принятия решения о необходимости создания системы и заканчивается в момент ее полного изъятия из эксплуатации.

Стандарты жизненного цикла определяют то, как создается, развертывается, применяется и ликвидируется система. Таким образом, жизненный цикл информационной системы охватывает все стадии и этапы ее создания, сопровождения и развития:

- предпроектный анализ (включая формирование функциональной и информационной моделей объекта, для которого предназначена информационная система);
- проектирование системы (включая разработку технического задания, эскизного и технического проектов);

- разработку системы (в том числе программирование и тестирование прикладных программ на основании проектных спецификаций подсистем, выделенных на стадии проектирования);
- интеграцию и сборку системы, проведение ее испытаний;
- эксплуатацию системы и ее сопровождение;
- развитие системы.

## **4.2 Анализ требований к системе. Истории пользователей**

Анализ требований к системе начинается с выявления всех заинтересованных сторон и определения ролей.

Одним из методов описания требований является метод написания историй пользователей (USER STORY) – это пользовательско-ориентированное описание целей, которые люди смогут достичь, используя ваш продукт, написанное повседневным языком.

### ***Формат:***

Я как \_\_\_\_\_, хочу \_\_\_\_\_, для того чтобы \_\_\_\_\_

Как <пользователь>, я могу <действие>, для того, чтобы <цель>

Основные вопросы User Stories:

- 1) ЗАЧЕМ хочет?
- 2) КТО хочет?
- 3) ЧТО хочет?

### ***Пример US:***

Как администратор компании, а хочу иметь возможность устанавливать заданные программы на удаленные рабочие столы, для того чтобы не тратить время на перемещение по офису и не отвлекать сотрудников от работы.

## **Рекомендации по выявлению действующих и заинтересованных лиц**

- 1) Найдите внешние сущности, с которыми разрабатываемая система должна взаимодействовать. Возможные кандидаты включают группы пользователей, которые обращаются к системе для выполнения своих задач или запускают первичные или вторичные функции системы, а также внешние устройства, программные средства и другие системы.
- 2) Дайте название и краткое описание каждому кандидату на действующее лицо. Включите область ответственности действующего лица и цели, которые действующее лицо пытается достичь при использовании системы. Исключайте тех кандидатов на действующее лицо, которые не имеют никаких целей.

Ответы на следующие вопросы должны помочь при определении действующих лиц:

- Кто будет поставлять, использовать или удалять информацию из системы?
- Кто будет использовать систему?
- Кто заинтересован в определенных характеристиках или услугах, предоставляемых системой?
- Кто будет поддерживать и управлять системой?
- Каковы внешние системные ресурсы?
- Какие другие системы будут взаимодействовать (нуждаются во взаимодействии) с разрабатываемой системой?

3) Внимательно изучите список заинтересованных лиц, которых вы определили. Не все заинтересованные лица будут действующими лицами (т.е. они не будут взаимодействовать с разрабатываемой системой непосредственно), но все равно этот список заинтересованных лиц весьма полезен для выявления кандидатов на действующие лица.

## **Рекомендации по формулировке требований**

### ***Типичное требование***

<Тип пользователя> должен иметь возможность <описание возможности>

### ***Требование с ограничениями и условиями***

<Тип пользователя> должен иметь возможность <описание возможности> с <показатель производительности> от <момент отсчета>, находясь в <условия эксплуатации>

Оператор должен иметь возможность произвести выстрел в течение 3 секунд с момента обнаружения цели радаром, находясь в сложных морских условиях.

### ***Требование - ограничение***

<Тип пользователя> не должен попадать под действие <соответствующее законодательство>

### ***Системное требование***

<Система> должна <выполняемая функция> не менее чем <количество> <объект> функционируя в <условия эксплуатации>

Телекоммуникационная система должна обеспечивать телефонную связь не менее чем с 10 абонентами, функционируя в условиях отсутствия источника внешнего электрического питания

### ***Периодическое требование***

<Система> должна <выполняемая функция> <объект> каждые <показатель производительности> <единица измерения>

Кофе-машина должна производить горячий напиток каждые 10 секунд

Имеет смысл немного остановиться и спросить, а зачем нужны шаблоны, в чем их собственно прелесть.

Каждому или большинству требований можно назначить шаблон. Шаблон имеет некоторый идентификатор, скажем номер. Т.е. все требования со схожим шаблоном будут иметь соответственно тот же самый номер шаблона и будут выглядеть структурно одинаковыми.

Что это дает:

- для быстрого и глобального изменения стиля формулировки достаточно внести изменения в шаблон.
- более простая и удобная технология обработки информации, составления запросов по отдельному атрибуту шаблона требований, сортировки, фильтрация, поиск.
- удобный способ защиты конфиденциальной информации - с помощью шаблона легко закрыть доступ к определенной части требования.

### **Примеры шаблонов требований с ограничением**

#### ***Производительность/возможность***

<Система> должна <выполняемая функция> <объект> не менее чем «производительность» раз в <единица измерения>

#### ***Производительность/возможность***

<Система> должна <выполняемая функция> <объект> типа <характеристика> в течение <производительность> <единица измерения>

#### ***Производительность/мощность***

<Система> должна <выполняемая функция> не менее чем <количество> <объект>

#### ***Производительность/своевременность***

<Система> должна <выполняемая функция> <объект> в течение <производительность> <единица измерения> с момента <событие>

#### ***Производительность/периодичность***

<Система> должна <выполняемая функция> не менее чем <количество> <объект> в течение <производительность> <единица измерения>

#### **Способность к взаимодействию/мощность**

<Система> должна <выполняемая функция> <объект> состоящий из не менее чем <производительность> <единица измерения> с <внешняя сущность>

#### **Устойчивость/периодичность**

<Система> должна <выполняемая функция> <объект> с <производительность> <единица измерения> каждые <производительность> <единица измерения>

#### **Окружение/работоспособность**

<Система> должна <выполняемая функция> <объект> функционируя в <условия эксплуатации>

### **Детализация требований**

Телекоммуникационная система должна поддерживать телефонную связь не менее чем с 10 абонентами (выделен показатель производительности) функционируя в условиях отсутствия внешнего источника электроэнергии (выделено ограничение)

### **Альтернативное представление детализации требований**

Функционируя в условиях отсутствия внешнего источника электроэнергии телекоммуникационная система должна поддерживать телефонную связь не менее чем с 10 абонентами  
телекоммуникационная система должна поддерживать радиосвязь не менее чем с 15 водителями скорой помощи

## **ПРИМЕР**

### **1) Цель системы**

photobank.tochka.net — система для фотографов, дизайнеров, которая бы позволила им хранить и обмениваться фотографиями, а также продавать их в runete.

Ожидается, что прибыль от системы будет достигаться за счет рекламы третьих компаний, также, возможно, за счет процента с продаж пользователями своих фотографий

### **2) Роли**

- Те, которые хранят и обмениваются своими фотографиями – «пользователи».
- Те, кто размещают свою рекламу, ориентированную на «пользователей» системы – «рекламодатели».
- Системе необходимы «администраторы», которые будут обеспечивать ее поддержку для блага других пользователей

### 3) Истории

- 1 Как пользователь я могу добавлять и хранить свои фотографии в системе, чтобы иметь возможность показать или продать их другим пользователям.
- 2 Как рекламодатель я могу помещать свою рекламу в системе, ориентированную на пользователей.
- 3 Как администратор я могу управлять фотографиями пользователей, так чтобы контент сайта был легальным.
4. Как гость я могу зарегистрироваться в системе, заполнив расширенный список полей для получения пользовательской учетной записи, позволяющей продавать фото.
5. Как гость я могу войти в систему под ранее созданной учетной записью на toshka.net , заполнив недостающие поля, для последующей работы.
6. Как пользователь я могу удалить свою учетную запись и перестать быть пользователем системы.
7. Как пользователь я могу изменить данные своей учетной записи.

**Примечание.** Краткую теорию по построению IDEF0-диаграмм см. в лекциях и лабораторных работах по дисциплинам «Теория информационных процессов и систем», «Информационный менеджмент».

## 5 Варианты заданий

Для выполнения индивидуальных заданий необходимо организовать рабочие группы по 2 человека в каждой. Выбрать вариант задания и утвердить его у преподавателя. Две группы не могут выполнять одно и то же задание. Можно предложить свой вариант задания.

По каждому из заданий определить состав основных и вспомогательных процессов ЖЦ системы, обосновать выбор.

Написать истории пользователей, определить приоритеты для отбора ключевых требований (не менее пяти историй для каждой роли).

Вам поручено создание системы:

1. сбора и классификации информации о файлах ПК «Файловый органайзер».
2. для организации работы куратора учебной группы «Куратор».
3. «Электронный учебник».
4. «Журнал преподавателя».

5. «Справочник по нечеткой математике».
6. «Терминал платежной системы».
7. «Ремонт сотовых телефонов».
8. Заказ билетов в театральной кассе.
9. Заказ блюд в интернет-ресторане
10. Заказ билетов в аэропорту.
11. Заказ книги в библиотеке.
12. Банкомат.
13. Запись на прием к врачу.
14. Организация проката автомобилей.
15. Сдача теста по дисциплине ЭТНМ.
16. Сдача недвижимости в аренду.
17. Отчет о сборе требований к разрабатываемой ИС.
18. Заказ такси.
19. Продажа железнодорожных билетов.
20. Сведения об успеваемости студента.
21. Регистрация курсовых работ.
22. Заказ товаров через каталог.
23. Электронный дневник школьника.
24. Учет успеваемости студентов.
25. Учет сдачи лабораторных работ.
26. Возврат товара в магазин.
27. Прокат велосипедов.
28. Регистрация авто на стоянке аэропорта.

## **6 Описание заданий**

### **1 «Файловый органайзер».**

- Система должна позволить организовать хранилище логических указателей на файлы с возможностью открытия выбранного файла в соответствующем приложении (например .doc в MS Word и т.д.).

- Особенности:

1) Организация папок, согласно заданной логике. Остальные файлы и папки отображаются «как есть». Под «заданной логикой» в данном контексте понимается некая метка (обозначение) для файла или папки. Например: метка «документы на студента» подразумевает логическое объединение ярлыков на файлы с такой меткой, расположенных в разных местах на диске без их физического перемещения (эта возможность дополнительная). Таким образом в результате получаем каталог «Документы на студента», в котором хранятся указатели (в виде ярлыков) на все выбранные файлы.

2) Один и тот же файл может иметь неограниченное количество меток.

3) Предусмотреть открытие файлов поставленной в соответствие программой.

4) Возможность сортировки по меткам.

- 5) Отбор по меткам (по одной и нескольким одновременно).
- 6) Поиск файлов по имени с указанием реального расположения файла на диске и логического, согласно его метке.

## **2 «Куратор».**

Система формирует на основе заданного шаблона письмо куратора родителям студента.

*Входные данные:*

- 1) ФИО родителей (возможность обращения к обоим родителям сразу или по отдельности к каждому).
  - 2) ФИО студента.
  - 3) Дата.
  - 4) Подпись. Например, «Декан ФПИ, профессор С.А. Курносов».
- Предусмотреть возможность задания нескольких подписей одновременно.
- 5) Текст шаблона. Шаблонов может быть сколько угодно.

*Выходные данные:* файл письма в формате MS Word, имя которого формируется следующим образом: ФИОСтудента\_темаПисьма. Например, ИвановС.Л.\_поощрениеДеканата2009. (Тема письма может браться автоматом из названия шаблона).

- Сохранение сформированных файлов происходит автоматически, т.о. предусмотреть задание пути к папке с подобными файлами.
- Предусмотреть открытие сформированного файла в MS Word.
- Предусмотреть множественное создание писем на основе выбранного шаблона.

## **3 «Электронный учебник».**

Система должна хранить все загруженные в нее учебники (ссылки на файлы). Система должна организовать переходы в документе формата MS Word, согласно выделенным темам. Предусмотреть создание пользовательских закладок и перехода по ним. Система должна сохранять последнюю прочитанную страницу и предлагать перейти к ней после запуска «Электронного учебника». Возможность создания шпаргалок – по желанию команды разработчиков.

## **4 «Журнал преподавателя»**

Спросить преподавателя.