



**НОСОВА Л.С.**

**ОСНОВЫ ПРОГРАММНОЙ  
ИНЖЕНЕРИИ**

Учебно-методическое пособие

ОУ ВО «Южно-Уральский институт управления и экономики»

**НОСОВА Л.С.**

**ОСНОВЫ ПРОГРАММНОЙ  
ИНЖЕНЕРИИ**

Учебно-методическое пособие

Челябинск

2015

**ББК 32.973я73**  
**УДК 004**  
**Л 33**

**Автор:** Носова Л.С. – кандидат педагогических наук, доцент кафедры информационных технологий и систем ЧОУ ВПО «Южно-Уральский институт управления и экономики».

**Рецензенты:** Т.Н. Лебедева – кандидат педагогических наук, доцент ОУ ВО «Южно-Уральский институт управления и экономики»;  
О.А. Дмитриева – кандидат педагогических наук, доцент кафедры информатики, информационных технологий и методики обучения информатике ФГБОУ ВПО «Челябинский государственный университет»;  
А.Г. Кириченко – кандидат технических наук, доцент ФГБОУ ВПО «Челябинский государственный университет».

**Носова, Л.С.**

Основы программной инженерии: Учебно-методическое пособие / Л.С. Носова. – Челябинск: Полиграф-Мастер, 2015. – 79 с.

Учебное пособие нацелено на формирование у обучающихся профессиональных компетенции по овладению области программной инженерии. Комплексная структура, представляющая сочетание теоретической информации, методических рекомендаций и практических занятий, позволяет студентам освоить последовательность шагов по созданию информационных систем.

Данное учебное пособие рекомендуется для самостоятельного изучения дисциплины «Основы программной инженерии», при подготовке к экзаменам и курсовой работе, а также в качестве методических рекомендаций для преподавателей.

Учебное пособие предназначается для студентов, обучающихся по направлению 230400.62 «Информационные системы и технологии» при изучении дисциплины «Основы программной инженерии», всех форм обучения.

**ISBN**

© ОУ ВО «Южно-Уральский институт  
управления и экономики», 2015  
© Носова Л.С., 2015

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ .....	4
Методические рекомендации.....	6
Лабораторная работа №1 .....	9
Лабораторная работа №2 .....	16
Лабораторная работа №3 .....	21
Лабораторная работа №4 .....	34
Лабораторная работа №5 .....	40
Лабораторная работа №6 .....	42
Лабораторная работа №7-8.....	45
Лабораторная работа №8 .....	52
Лабораторная работа №9 .....	57
Лабораторная работа №14-16.....	65
ПРИЛОЖЕНИЕ .....	75
СПИСОК ЛИТЕРАТУРЫ.....	78

## ВВЕДЕНИЕ

В настоящее время создание программных продуктов, с одной стороны, остается сложной задачей на стыке искусства, ремесла и логики, с другой стороны – многие процессы здесь автоматизированы и подкреплены соответствующими инструментами. Оставаясь прикладной областью, программная объединяет в себе множество практикоориентированных компонентов: тестирование программного обеспечения, программирование, проектирование и др. Основные слагаемые успеха инженеров программного обеспечения не являются секретом и лежат на поверхности: они придерживаться установленного плана, обеспечить качество продукта, не выходя при этом за рамки бюджета. Однако не все студенты в ходе освоения дисциплины учитывают, что помимо непосредственной разработки программных систем и их сопровождения, им необходимо анализировать требования заказчика, разрабатывать спецификации, реализовать сам проект, интегрировать его с другими, аттестовать и т.п.

Данное учебное пособие составлено на основе многолетнего опыта преподавания дисциплины «Основы программной инженерии» в ОУ ВО «Южно-Уральский институт управления и экономики» в рамках подготовки по направлению бакалавриата 09.03.02 «Информационные системы и технологии». С другой стороны, оно может быть интересно и полезно и для студентов других направлений и специальностей, нуждающихся в практических рекомендациях по разработке программных продуктов, а также для коллег-преподавателей из сферы высшего и среднего профессионального образования.

Учебное пособие нацелено на оказание помощь студентам в формировании их общекультурных и профессиональных компетенций в соответствии с федеральным государственным образовательным стандартом высшего образования: 09.03.02 «Информационные системы и технологии». Компетенции, формируемые в освоения дисциплины, представлены в таблице 1.

Таблица 1– Формируемые компетенции

<b>Код компетенции</b>	<b>Наименование компетенции</b>
ПК-2	способность проводить техническое проектирование
ПК-33	способность составлять инструкции по эксплуатации информационных систем

## Методические рекомендации для преподавателей

Дисциплина «Основы программной инженерии» осуществляет:

- подготовку выпускников к проектно-технологической деятельности в области создания компонентов программных комплексов и баз данных, автоматизации технологических процессов с использованием современных инструментальных средств и технологий программирования;
- подготовку выпускников к комплексным инженерным исследованиям для решения задач, связанных с разработкой программных средств, к работе по созданию программного обеспечения в проектных группах;
- обучение методам командной работы.

После изучения данной дисциплины выпускник должен решать следующие профессиональные задачи в соответствии с видами профессиональной деятельности.

Проектно-конструкторская деятельность:

- предпроектное обследование (инжиниринг) объекта проектирования, системный анализ предметной области, их взаимосвязей;
- техническое проектирование (реинжиниринг);
- рабочее проектирование;
- выбор исходных данных для проектирования;
- моделирование процессов и систем;
- сертификация проекта по стандартам качества;
- расчет обеспечения условий безопасной жизнедеятельности.

Проектно-технологическая деятельность:

- проектирование базовых и прикладных информационных технологий.

В настоящее время ситуация на рынке труда такова, что при устройстве на работу от выпускника требуется опыт разработки программного обеспечения. Выйдя из стен высших учебных заведений, вчерашние студенты сталкиваются с рядом проблем: у них нет опыта создания программ в реальных условиях, они не умеют работать в команде. Возникает неуверенность в своих возможностях и перспективах.

Конечно, при подготовке будущих инженеров преподаватели применяют различные подходы к повышению качества уровня знаний,

умений и навыков студентов. Например, это может быть рейтинговая (балльно-рейтинговая) система оценивания студентов, формулирование заданий на курсовой проект на основе реальных заказов потребителей, конкурсы на самый лучший курсовой проект или квалификационную работу и т. п.

Однако это не всегда готовит студента к реальной разработке. В связи с этим в данном пособии предложен подход к обучению проектированию программного обеспечения (ПО) на основе методологии разработки ПО Microsoft Solutions Framework (MSF). Освоение данного продукта приблизит студента к реальному процессу разработки программ, позволит получить навыки работы в команде.

Методология MSF предложена корпорацией Microsoft и опирается на практический опыт компании. В ней описывается последовательность шагов по управлению людьми и рабочими процессами в ходе разработки программного решения.

Для учебного процесса данная методология позволит четко регламентировать вехи (контрольные точки проекта) и тем самым реализовать контроль над выполнением проектов. Кроме того, фазы, представленные в методологии, соответствуют фазам, которые проходят студенты в процессе выполнения заданий, соответственно они смогут оценить время, необходимое для выполнения работы в целом.

MSF предлагает следующие фазы работы над проектом:

1. Выработка концепции проекта. На этапе происходит осмысление полученного задания студентом, уточнение и предварительный поиск способов решения. В качестве промежуточного результата формируется концепция проекта.

2. Проектирование. На этом этапе осуществляется проектирование продукта различными методами сбора информации. В качестве подитога формируются сценарии использования, описание логического и физического дизайна средствами диаграмм UML.

3. Реализация. Результатом данной технологической фазы являются программный код и beta-версия ПО.

4. Тестирование. На этапе осуществляется тестирование разработанного ПО: функциональное, нагрузочное, стресс-тестирование. В



результате прохождения этапа исполнитель создает спецификацию тестирования, план внедрения и сам программный продукт.

5. Подведение итогов. На этапе осуществляется сбор всей документации, полученной на предыдущих этапах, а также организуется защита проекта перед преподавателем и/или комиссией. Финальным результатом являются оценки студентов по дисциплине в экзаменационной ведомости.

Стоит оговориться, что при выборе данного подхода увеличивается нагрузка на преподавателя, т. к. ему необходимо выступать еще и в роли заказчика, а значит быть на связи со студентом в любое время. Эти вопросы решаются, например, посредством электронной почты. Кроме того, можно использовать специальные программы-органайзеры работы в команде, например, онлайн-сервис Битрикс24.

В данном пособии представлены лабораторные работы для организации работы студентов в рамках первых двух фаз и последнего этапа – подведения итогов в виде конвейера проектов.

В качестве проектов, которые будут разрабатывать студенты в течение курса, могут выступать их курсовые работы или реальные заказы образовательной организации.

## **Лабораторная работа №1**

### **Методологии управления ИТ-проектами**

**Цель:** знакомство с методологиями управления ИТ-проектами.

#### **Теоретические вопросы**

В настоящее время управление проектами имеет свою методологию и основывается на определенных стандартах. В основе таких методов лежат методики сетевого планирования, которые появились США в конце 1950-х гг. При этом методики управления проектами широко распространились не только в странах с рыночной экономикой, но и в странах с т. н. «плановой» экономикой. Они начали использоваться в строительстве, что послужило основой их распространения в других отраслях и появлению методов проектного управления.

Примером здесь может служить объединение усилий фирмы «Дюпон» и фирмы «Ремингтон Рэнд» для составления плана графика комплексных работ по модернизации заводов «Дюпон» с помощью вычислительной машины Univac в 1956 г. Результатом стало создание рационального и простого метода описания проекта на ЭВМ – метода критического пути (CPM – Critical Path Method).

Практически параллельно и независимо был создан метод анализа и оценки программ PERT (Program Evaluation and Review Technique) в военно-морских силах США. Данный метод разрабатывался корпорацией «Локхид» и консалтинговой фирмой «Буз, Аллен энд Гамильтон» для ракетной системы «Поларис». Проект включал 3800 основных подрядчиков и состоял из 60 000 операций. Благодаря использованию данного метода проект стал успешным и завершился на два года раньше срока. После наглядного успеха метод PERT стал использоваться в вооруженных силах США для планирования проектов.

**Задание:** найдите примеры других крупных успешных проектов прошлого века, использующих для своего управления ЭВМ.

Учтите, что с одной стороны, применение информационных технологий для планирования и управления проектами давало в те времена значительное преимущество, а с другой стороны, первые компьютеры были дорогостоящими и оставались доступными только крупным компаниям. Все

изменилось с появлением персональных компьютеров, теперь ЭВМ стала рабочим инструментом для руководителей более широкого круга.

Система управления проектами постоянно развивалась и стала самостоятельной областью профессиональной деятельности. В итоге были созданы унифицированные методологии, инструментарии, механизмы, стандарты, доступные и для проектов в ИТ-сфере. Например, на сегодняшний день существует единая Международная ассоциация управления проектами – IPMA (International Project Management Association) с центром в г. Цюрих (Швейцария).

Из наиболее распространенных можно отметить процессную модель, используемую в документах методологических основ управления проектами – Project Management Body of Knowledge (PMBOK) Американского института управления проектами (PMI). Данный документ признается международным стандартом де-факто. Кроме того, стандарт ISO 10006:1997 придал ряду наиболее важных положений PMBOK статус стандарта де-юре.

В 2014 г. вышло пятое издание PMBOK, содержащее указания на 589 страницах. Все положения представлены на сайте <http://www.pmi.org/default.aspx> [6].

Данные Международной ассоциации управления проектами (IPMA) говорят о том, что использование современных методологий управления проектами экономит около 20–30% времени и около 15–20% средств на осуществление проектов.

Все методологии (еще их называют моделями, методиками) разработки программного обеспечения классифицируют по «весу», т. е. по количеству формализованных процессов и детальности их регламентации. Следовательно, чем больше процессов документировано, чем более детально описана методология, тем больше будет ее «вес».

1. Тяжеловесные методологии:

– ГОСТ 19 «Единая система программной документации» и ГОСТ 34 «Стандарты на разработку и сопровождение автоматизированных систем» ориентированы на последовательный подход к разработке программного обеспечения;

– Capability Maturity Model for Software (SW-CMM) определяет пять уровней «зрелости проекта»;

- Rational Unified Process (RUP) – итеративная модель разработки;
  - Microsoft Solutions Framework (MSF) – база знаний компании Microsoft по разработке программ;
  - Personal Software Process – модель определяет требования к компетенциям разработчика. Team Software Process – модель ориентирует на самоуправляемые команды от 3 до 20 разработчиков.
  - и др.
2. Легковесные или agile-методики:
- eXtreme Programming или XP – экстремальное программирование, предлагающее 12 инженерных практик;
  - Crystal Clear – семейство методологий, определяют необходимую степень формализации процесса разработки в зависимости от количества участников и критичности задач [13];
  - Feature Driven Development (FDD) – функционально-ориентированная разработка;
  - OpenUP – итеративно-инкрементальный метод разработки программ, позиционируется как легкий и гибкий вариант тяжеловесной методологии RUP;
  - Scrum – управление разработкой информационных систем с высокой степенью неопределенности;
  - Kanban – методология «бережливого производства»;
  - и др.

Результаты исследования Agile Survey о популярности гибких методологий представлены на рис. 1 [11].

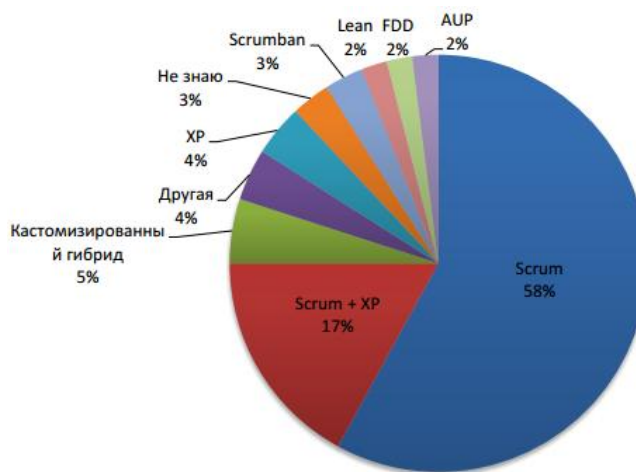


Рис. 1. Популярность гибких методологий

Легковесные методологии Agile появились сравнительно недавно. В феврале 2001 г. 17 специалистов (консультантов и практиков) провели семинар, на котором сформулировали основные принципы гибкой разработки ПО – Agile Manifesto – манифест гибкой разработки. Он переведен на многие языки мира и доступен на сайте <http://agilemanifesto.org/iso/ru/> [12].

Идея всех гибких (легковесных) методологий состоит в том, что применяемый в разработке ПО процесс должен быть адаптивным, ориентированным на людей и их взаимодействие. По сути это скорее набор практик, чем методология.

На том же семинаре было предложено новое название таких методологий – гибкая разработка Agile Software Development.

При выборе модели управления проектом можно ориентироваться на следующую таблицу (таблица 2).

Таблица 2 – Выбор методологии управления проектом

Вес модели	Достоинства	Недостатки
Тяжеловесные	<ul style="list-style-type: none"> <li>– процессы рассчитаны на среднюю квалификацию исполнителей;</li> <li>– большая специализация исполнителей;</li> <li>– низкие требования к стабильности команды;</li> <li>– отсутствуют ограничения по объему и сложности выполняемых проектов.</li> </ul>	<ul style="list-style-type: none"> <li>– требуют существенной управленческой надстройки;</li> <li>– длительные стадии анализа и проектирования;</li> <li>– формализованные коммуникации.</li> </ul>

Продолжение таблицы 2

Легковесные (гибкие)	<ul style="list-style-type: none"> <li>– снижение непроизводительных расходов, связанных с управлением проектом, рисками, изменениями, конфигурациями;</li> <li>– упрощение стадий анализа и проектирования, основной упор на разработку функциональности, совмещение ролей;</li> <li>– неформальные коммуникации.</li> </ul>	<ul style="list-style-type: none"> <li>– эффективность сильно зависит от индивидуальных способностей, требуют более квалифицированной, универсальной и стабильной команды;</li> <li>– объем и сложность выполняемых проектов ограничены.</li> </ul>
----------------------	---	---

А. Коуберн, один из авторов «Манифеста», провел анализ ИТ-проектов за последние 20 лет [4], выполненных на основании разных моделей управления: от облегченных, гибких до тяжелых (СММ-5). Данный анализ показал отсутствие корреляции между успехом или провалом проектов и выбранными моделями разработки, применяемыми в проектах.

А. Коуберн сделал также вывод о том, что:

1. У каждого проекта должна быть своя модель процесса разработки.
2. У каждой модели – свое время.

Для разработчиков это означает, что не существует единственного правильного процесса разработки. В каждом новом проекте процесс должен определяться каждый раз заново по «закону четырех П» (рис. 2).

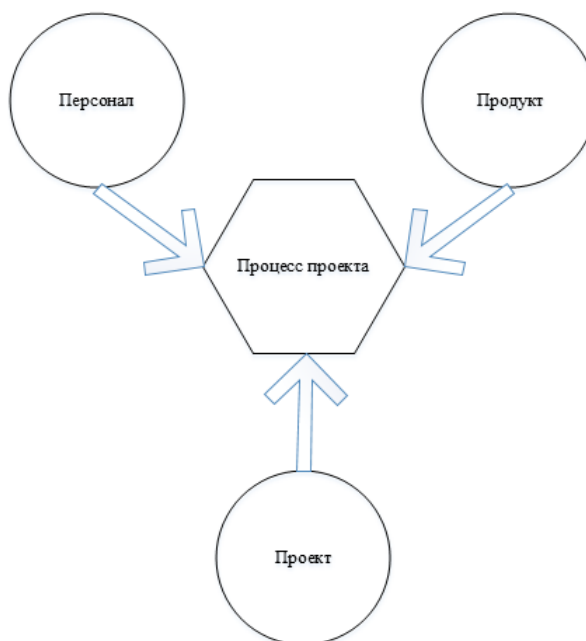


Рис. 2. Закон четырех П

## Ход работы

**Задание 1.** С помощью поиска в сети Интернет найдите информацию о современных методологиях управления ИТ-проектами. Представьте основания для их классификации. Для каждого основания приведите примеры методологий.

**Задание 2.** Из полученного списка тяжеловесных методологий управления ИТ-проектами выберите один. Проведите исследование методологии. Результат представьте в таблице (таблица 3).

Таблица 3 – Особенности методики

Характеристика	Описание
Полное название методологии	
Авторы	
История возникновения	
Страна появления	
Основные принципы, подходы	
Имеются ли программные средства реализации методологии, какие?	
Используется ли в настоящее время	
Примеры успешных проектов, реализованных с помощью данной методологии	

**Задание 3.** Из полученного списка легковесных (agile) методологий управления ИТ-проектами выберите один. Проведите исследование методологии. Результат представьте в таблице (таблица 3).

**Задание 4.** Выберите любую из проанализированных методологий. Создайте о ней презентацию на 10-15 слайдов. Выступите в группе, будьте готовы ответить на вопросы.

### Контрольные вопросы:

1. Что такое методология управления ИТ-проектом?
2. Какие виды методологий вы знаете?
3. В чем особенности тяжеловесных и легковесных методологий управления?
4. Приведите примеры методологий, используемых для разработки ИТ-проектов.

**По завершении занятия студент должен:**

1. Знать понятие методологии управления ИТ-проектами, их виды.
2. Приводить примеры различных методологий.
3. Перечислять преимущества тяжеловесных и легковесных методологий.
4. Осуществлять выбор методологий управления при работе над ИТ-проектом.



## Лабораторная работа №2

### Анализ рисков

**Цель:** осуществить анализ рисков методами «Матрица компромиссов» и «Таблица анализа рисков».

#### Теоретические вопросы

Для более глубокого понимания компромиссов программного проекта, часто бывает очень полезно изобразить заказчику зависимость основных характеристик проекта в виде треугольника компромиссов (рис. 3).



Рис. 3. Треугольник компромиссов

Утвержденное равновесие с заказчиком достигается в том случае, если исполнитель с учетом запрашиваемых параметров назвал и зафиксировал время (сроки) и ресурсы (смету). Следует учитывать, что любое изменение одной из сторон треугольника обязательно влечет изменение на двух оставшихся.

Для эффективного достижения компромиссов в течение всего жизненного цикла программного проекта на начальных этапах следует выявить и зафиксировать приоритет факторов (ресурсы, время, возможности). Если один из факторов фиксируется как неизменный, то влиять на него в течение проекта практически невозможно. Второй фактор согласовывается по следующему принципу: он будет обладать некоторым приоритетом в случае необходимости достижения компромиссов. Последний фактор просто принимается в соответствии с первыми двумя. Матрица компромиссов представлена в табл. 4

Таблица 4 – Матрица компромиссов

	Фиксируется (Зафиксировано)	Согласовывается (Определено)	Принимается (Корректируемо)
Ресурсы			
Время (график)			
Возможности (набор функций программы)			

Традиционная модель управления проектами подразумевает четкую формулировку требований на начальном этапе проекта, разработку на основании технического задания. Подход компромиссов основывается на принципе изменяющихся проектных условий. Разработчикам необходимо проявлять гибкость и в любой момент времени быть готовым к изменениям и рискам. Методология Microsoft Solution Framework предлагает вычленять возможные риски и анализировать их, чтобы ими можно было управлять.

Риском называется проблема, которая еще не возникла. В свою очередь, проблемой называют риск, который уже материализовался. Причиной возникновения любых рисков являются неопределенности в программном проекте. Следовательно, нужно стремиться к выявлению рисков. Иногда это осуществляется методом мозгового штурма, опроса экспертов, метода Дельфи и др. или на опыте предыдущих проектов, опыте других разработчиков.

Риски делят на два вида:

- 1) известные – риски, которые определены, оценены, для них возможно планирование и анализ;
- 2) неизвестные – они не могут быть заранее идентифицированы или спрогнозированы.

Б. Бозм приводит список 10 наиболее распространенных рисков программного проекта [1]:

- 1) дефицит специалистов;
- 2) нереалистичные сроки и бюджет;
- 3) реализация несоответствующей функциональности;
- 4) разработка неправильного пользовательского интерфейса;
- 5) «золотая сервировка», перфекционизм, ненужная оптимизация и оттачивание деталей;

- б) непрекращающийся поток изменений;
- 7) нехватка информации о внешних компонентах, определяющих окружение системы или вовлеченных в интеграцию;
- 8) недостатки в работах, выполняемых внешними (по отношению к проекту) ресурсами;
- 9) недостаточная производительность получаемой системы;
- 10) «разрыв» в квалификации специалистов разных областей знаний.

Смысл того, чтобы описывать риски и проводить их анализ сводится к как можно более раннему выявлению этих рисков, проведению комплекса предупреждающих мероприятий. Все это позволит избежать трудноразрешимых проблем уже на этапе реализации проекта.

Все риски оцениваются в таблице (таблица 5).

Таблица 5 – Анализ рисков

№ п/п	Риск	Последствия наступления риска	Меры по предотвращению	Меры по минимизации

Поля таблицы означают следующее.

1. Под риском понимается событие, которое следует однозначно и конкретно сформулировать, например, погодные условия.
2. В третьем графе оцениваются отрицательные последствия материализации этого события.
3. Под мерами по предотвращению подразумеваются возможные действия команды во избежание материализации риска.
4. Меры по минимизации – это действия команды для сокращения потерь на случай, если событие все-таки произошло.

## Ход работы

### Работа в группе

**Задание 1.** Создать матрицу компромиссов для проекта «Квалификационная работа» (таблица 4).

**Задание 2.** Разделиться на две группы и методом мозгового штурма определить список рисков для проекта «Квалификационная работа» (таблица б).

Таблица 6 – Список рисков

№ п/п	Риск

После обсуждения объединить результаты групп и методом экспертных оценок произвести оценку полученного списка рисков (таблица 7).

Таблица 7 – Экспертная оценка

Риск	Эксперт 1	Эксперт ...	Эксперт N	Рейтинг риска

**Задание 3.** Из полученного списка рисков (таблица 6) выбрать пять с самым высоким рейтингом. Составить таблицу «Анализ рисков проекта «Квалификационная работа» (таблица 5).

**Задание 4. Индивидуальная работа**

Для выбранного варианта информационной системы (Приложение):

- заполнить матрицу компромиссов (Таблица 4);
- написать список рисков (Таблица 6);
- повести анализ рисков (Таблица 5).

**Контрольные вопросы**

1. Что такое треугольник компромиссов?
2. Когда и для чего используется треугольник компромиссов?
3. К чему приводит изменения одной из сторон треугольника компромиссов?
4. Приведите пример заполненной матрицы компромиссов, например, для проекта челябинского метро.
5. Что такое риск?
6. Назовите виды рисков.
7. Приведите наиболее распространенные риски для программного проекта.
8. Для чего управлять рисками?
9. Как управлять рисками?
10. Поясните поля таблицы анализа рисков.

**По завершении занятия студент должен:**

1. Знать, что такое риски.
2. Иметь представление о видах рисков.
3. Знать методы анализа рисков.
4. Заполнять матрицу компромиссов для конкретного программного проекта.
5. Пояснять методологию анализа рисков, указывать ее назначение.
6. Приводить примеры рисков.
7. Осуществлять анализ рисков с помощью таблицы анализа рисков.

## **Лабораторная работа №3**

### **Сравнительный анализ информационных систем**

**Цель:** провести анализ аналогов – информационных систем из одной предметной области – для выявления требований к разрабатываемому программному продукту.

#### **Теоретические вопросы**

Под проектированием понимают процесс создания проекта, т. е. прототипа, прообраза предполагаемого или возможного объекта или состояния, а также комплекта документации к нему.

От специфического для машиностроения, строительства и других отраслей науки и техники понятия «проект» (англ. design) в значении «проектная документация» следует отличать используемое в контексте менеджмента понятие «проект» (англ. project, от лат. projectus – брошенный вперед, выступающий) в значении «некоторая задача с определёнными исходными данными и требуемыми результатами (целями), обуславливающими способ её решения», «программа», «комплекс работ» и т. п. [10].

Проект (от лат. projectus – брошенный вперед, выступающий, выдающийся вперед) – это уникальный процесс, состоящий из совокупности скоординированных и управляемых видов деятельности с начальной и конечной датами, предпринятый для достижения цели, соответствующей конкретным требованиям, включающий ограничения по срокам, стоимости и ресурсам [3].

С точки зрения системного подхода проект может рассматриваться как процесс перехода из исходного состояния в конечное — результат при участии ряда ограничений и механизмов.

Проект – некоторая задача с определенными исходными данными и требуемыми результатами (целями), обуславливающими способ ее решения. Проект включает в себя замысел (проблему), средства его реализации (решения проблемы) и получаемые в процессе реализации результаты [8].

В общем смысле слова проект – это все то, что изменяет каким-то образом наш мир, делает его лучше. Решающую роль для проекта имеют три составляющих: цель, бюджет и время.

В свою очередь, от любой задачи проект отличается однократностью действий. Серийное производство не имеет ограничения во времени, проект ограничен по срокам реализации ровно настолько, сколько требуется для достижения результата.

Можно выделить следующие критерии проекта:

- новизна;
- наличие четкой целевой установки;
- начало и окончание проекта четко определены во времени;
- комплексность поставленной цели; проект в основном включает в

себя связанные между собой и зависимые друг от друга задачи и подзадачи разного уровня;

- участие нескольких специалистов для достижения цели.

Признаки «не проекта»:

- цель однозначно не определена, неконкретна или недостижима;
- сроки не определены, нереальны для достижения цели;
- результат не уникален.

Классифицировать проекты можно следующим образом:

- по уровню проекта: проект, программа, система;
- по масштабу проекта: малый, средний, мегапроект;
- по сложности: простой, организационно-сложный, технически сложный, ресурсно-сложный, комплексно-сложный;
- по срокам реализации: краткосрочный, средний, мегапроект.

Существуют и другие основания.

Как показывает статистика компании The Standish Group [9], в мире всегда имеются успешные проекты (достигли цели без перерасхода бюджета и сроков реализации), спорные (достигли цели с перерасходом бюджета и/или сроков реализации) и провальные (не достигли цели) (рис. 3).

	2004	2006	2008	2010	2012	2013
УСПЕШНЫЕ	29%	35%	32%	37%	39%	36%
ПРОВАЛЬНЫЕ	18%	19%	24%	21%	18%	16%
СПОРНЫЕ	53%	46%	44%	42%	43%	48%

Рис. 3. Статистика успешности проектов

Данные по успешности ИТ-проектов компании The Standish Group представлены на рисунке 4.



Рис. 4. Статистика успешности ИТ-проектов

Обычно в качестве причин неудач называются следующие:

- неполные требования;
- низкая степень вовлечения заказчика и конечных пользователей в процесс разработки;
- недостаточное обеспечение ресурсами;
- недостаток планирования и др.

### Ход работы

**Задание 1.** Осуществить в сети Интернет поиск готовых информационных систем, решающих задачу из предметной области, выбранную вами в соответствии с вариантом (Приложение 1). Представить результат в виде списка информационных систем (таблица 7).

Таблица 7 – Программные продукты из предметной области

№ п/п	Название продукта	Название фирмы	Требования к системе	Возможности	Стоимость

**Задание 2.** Из представленной выше таблицы выбрать три программных продукта и провести их сравнительный анализ. Результат: характеристики продуктов, представленные в таблице 8.

Таблица 8 – Сравнение программных продуктов

№ п/п	Список характеристик	Название продукта №1	Название продукта №2	Название продукта №3
		Представлена характеристика или нет	Представлена характеристика или нет	Представлена характеристика или нет

**Задание 3.** На основании таблиц сделать вывод, какой должна быть ваша информационная система, чтобы учитывать все достоинства и



недостатки готовых программных продуктов. Результат представить в виде списка отличий.

**Задание 4.** Для вашей системы составить список тех пользователей, которые будут иметь дело с разрабатываемым программным продуктом.

**Задание 5.** Для каждого пользователя определить список его возможностей в вашей информационной системе (описание должно быть сделано на языке, понятном пользователю!).

**Задание 6.** Обсудите в группе результаты работы. Назначьте ответственных за ведение документации.

### **Контрольные вопросы:**

1. Что такое проект, программный проект, проектирование?
2. Чем отличается задача от проекта, приведите примеры.
3. Назовите основания для классификации проектов.
4. Каковы критерии успешности проектов?
5. С какой целью проводится анализ аналогов разрабатываемого программного продукта.
6. Для чего составляется список пользователей программного продукта?
7. Кто такие заинтересованные лица проекта?

### **По завершении занятия студент должен:**

1. Знать понятие проекта, программного проекта, проектирования.
2. Иметь представление о критериях проекта и его отличия от задачи.
3. Знать классификацию проектов.
4. Иметь представление об успешности проекта.
5. Осуществлять анализ программных продуктов из предметной области с целью выявления требований к разрабатываемому программному проекту.
6. Формулировать список заинтересованных лиц и будущих пользователей программного проекта.

## **Лабораторная работа №4**

### **Временное планирование проекта. Диаграмма Ганта**

**Цель:** осуществить временное планирование программного проекта.

#### **Теоретические вопросы**

Как показывает практика управления проектами, использование методик планирования и контроля за ходом работ не только позволяет ускорить выполнение проекта, но и в значительной мере уменьшает затраты на реализацию программы.

После того как определена цель проекта, необходимо составить список работ для выполнения. Менеджеры проекта рекомендуют составить и придерживаться плана проекта. Этот документ четко указывает контрольные точки проекта и основные действия, необходимые для достижения цели проекта. Кроме того, он определяет дату каждой контрольной точки (завершения главных действий по достижению целей) и ответственных за каждое действие. План составляется на начальном этапе, одобряется командой проектировщиков.

План может быть составлен на примере схожих проектов, или это может быть просто список действий в формальном виде. Необходимо всегда записывать фактические даты завершения каждого этапа и/или действия. Если дата отличается от плана, в этом случае нужно провести корректировку.

При включении необходимых пунктов в план следует:

- применять уникальные идентификаторы (ID), которые можно использовать в случае, если необходимы обновления;
- давать названия задачам;
- указывать начало выполнения действия;
- указывать дату завершения выполнения;
- записывать фактическую дату завершения выполнения действия;
- любую задачу завершать до того, как будет начата следующая;
- указывать ответственного за выполнение (хозяина задачи);
- указывать процент завершенности каждого действия.

Обычно назначается ответственный для осуществления контроля над выполнением плана и внесения изменений в него.

Одним из способов контроля над временем, планирования проекта может служить диаграмма Ганта.

**Диаграмма Ганта** (Gantt chart, ленточная диаграмма, график Ганта) – тип столбчатых диаграмм, ставших популярным в проектном менеджменте. Используется для наглядного представления плана проекта или графика работ [7].

Г.Л. Гант (1861 – 1919) изучал менеджмент на примере постройки кораблей. Он предложил свой вариант диаграммы во время Первой мировой войны, когда потребовалось отслеживать ход строительства больших трансконтинентальных океанских лайнеров. Такая диаграмма состояла из отрезков (задач) и точек, т. н. «завершающих задач», или «вех». И выступала средством представления длительности и последовательности задач в проекте.

Идея планирования состоит в том, что его главным ресурсом является время. В связи с этим основой для принятия управленческих решений является сравнение запланированного состояния работ и фактического. На диаграммах по горизонтали указываются интервалы времени, по вертикали – операции, работы или оборудование. Горизонтальные отрезки иллюстрируют длительность выполнения работ. Выбрав по горизонтали текущий момент времени и получив оперативную информацию о ходе проекта, можно сопоставить состояние дел по факту и состояние дел по плану.

На рис. 5 представлен пример диаграммы Ганта.

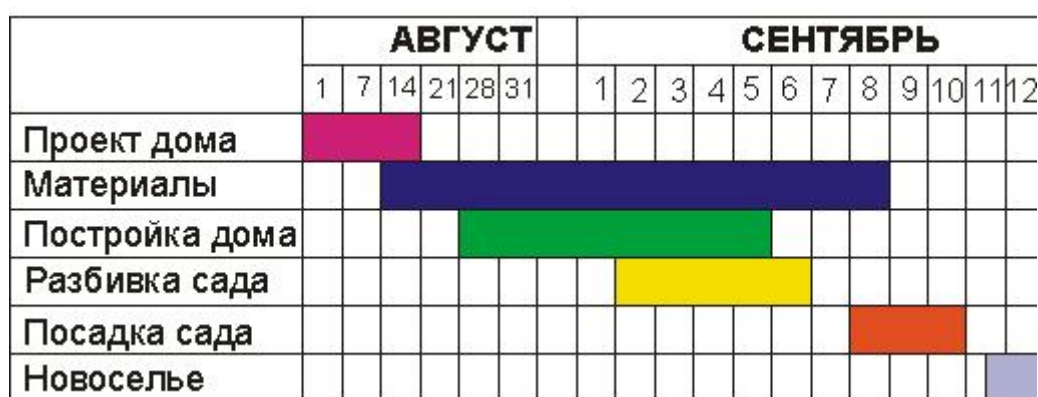


Рис. 5. Пример диаграммы Ганта

Диаграмма Ганта позволяет:

- визуально оценить последовательность задач, их относительную длительность и протяженность проекта в целом;
- сравнить планируемый и реальный ход выполнения задач;

– детально проанализировать реальный ход выполнения задач (например, задача выполнялась, была приостановлена, возвращалась на доработку и т. д.).

Методика впервые была представлена в 1910 г. Со временем диаграмма Ганта стала главным инструментом, используемым в календарном планировании и контроле. В 1990-х гг. методика была усовершенствована: для описания зависимостей между задачами были добавлены связи.

Рассмотрим типы связей (рис. 6):

1. **Финиш-Старт.** Данная связь означает, что операция В не может начаться до завершения операции А, или дата окончания операции А определяет дату начала операции В. Например, надо сначала написать диплом, а потом его можно защищать.

2. **Финиш-Финиш** – операция В должна закончиться не раньше операции А, или дата окончания операции А определяет дату окончания операции В. Например, если Вы пишете клиент-серверное приложение (операция А) и для его отладки вы брали в аренду сервер (операция В), то отладка должна завершиться к сроку окончания аренды сервера.

3. **Старт-Старт** – операция В начинается не раньше операции А, или дата начала операции А определяет дату начала операции В. Например, операции печати диплома тесно связана с покупкой бумаги и задачи должны решаться практически одновременно.

4. **Старт-Финиш** – операция В не может закончиться пока не начнется операция А, или дата начала операции А определяет дату окончания операции В. Время, на которое запланирована защита диплома, определяет, когда должны завершиться предзащиты.

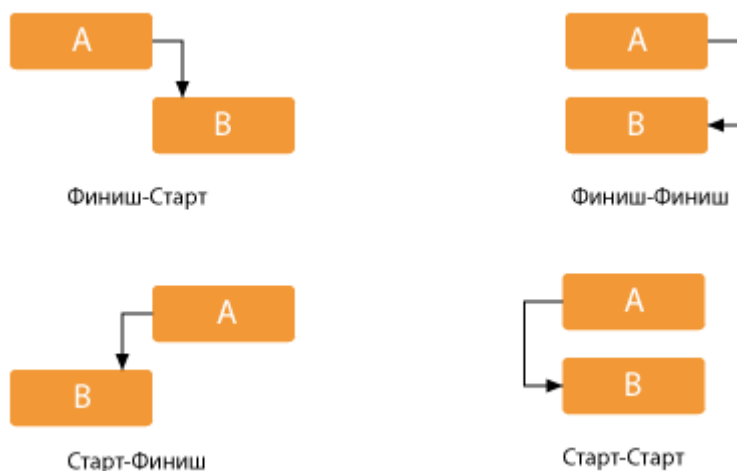


Рис. 6. Типы связей на диаграмме Ганта

Однако диаграммы имеют недостатки: с их помощью сложно планировать многовариантные взаимосвязанные цепочки работ (для строительных, военных, государственных проектов и на производстве). Для таких задач предложены методы сетевого планирования, или методы выбора «критического пути», разработанные военным ведомством США в 1950-х гг. Диаграммы удобно применять только для одного ресурса – времени. Если учитывать еще несколько ресурсов, то диаграммы Ганта надо воспринимать в объемном виде, что имеет смысл для визуальной интерпретации планов с одной стороны, но затрудняет их анализ с другой. Для управления проектами обычно используются другие, более мощные средства – метод «критического пути», (Critical Path Method, CPM) и метод PERT (Program Evaluation Review Technique).

Программно диаграммы могут быть реализованы различными средствами, в том числе и онлайн. Рассмотрим примеры (рис. 7-11).

Внешний вид диаграмм, созданных с помощью Microsoft Project (рис. 7).

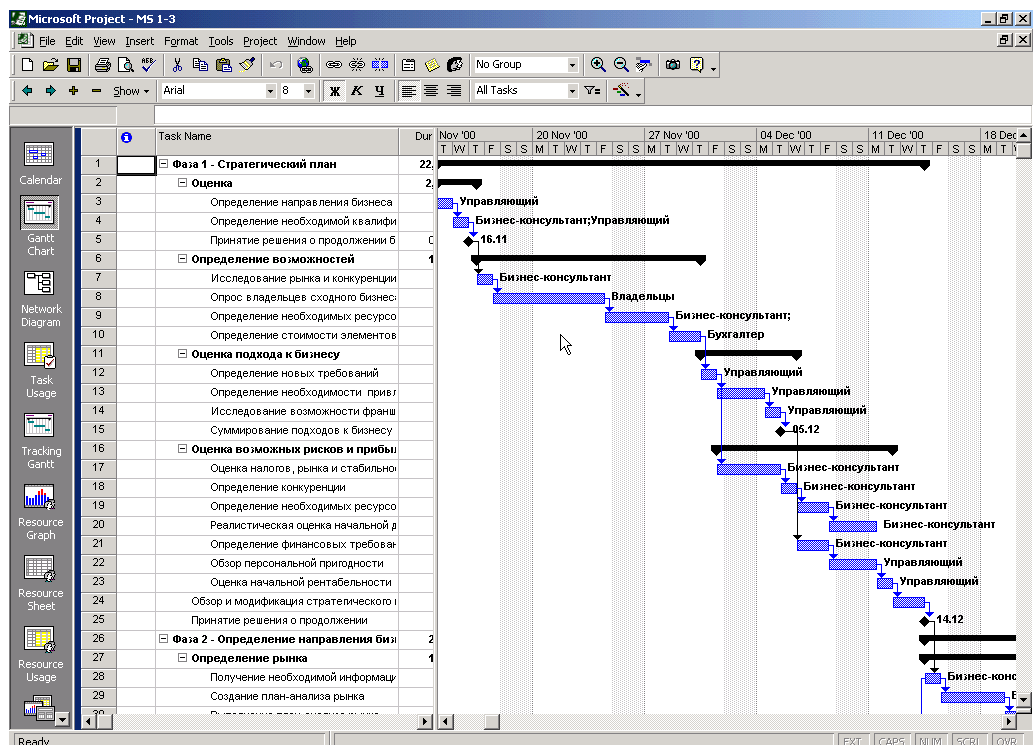


Рис. 7. Диаграмма Ганта в Microsoft Project

OpenProj – программное обеспечение для управления проектами, кроссплатформенное (рис. 8).

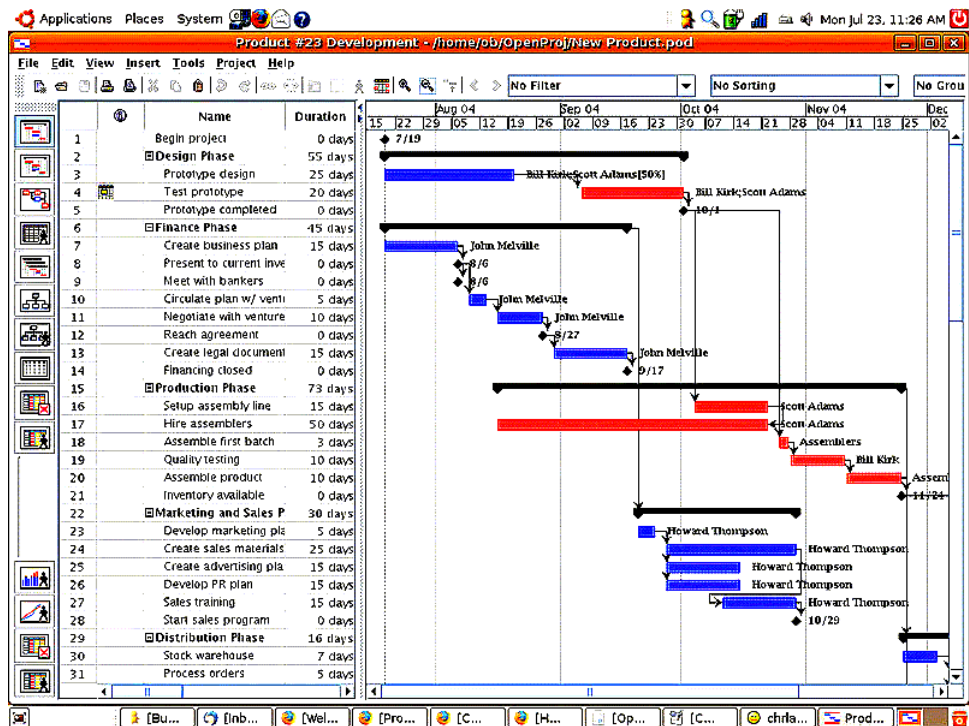


Рис. 8. Диаграмма Ганта в OpenProj

Внешний вид диаграмм в программе @Управление (рис. 9).

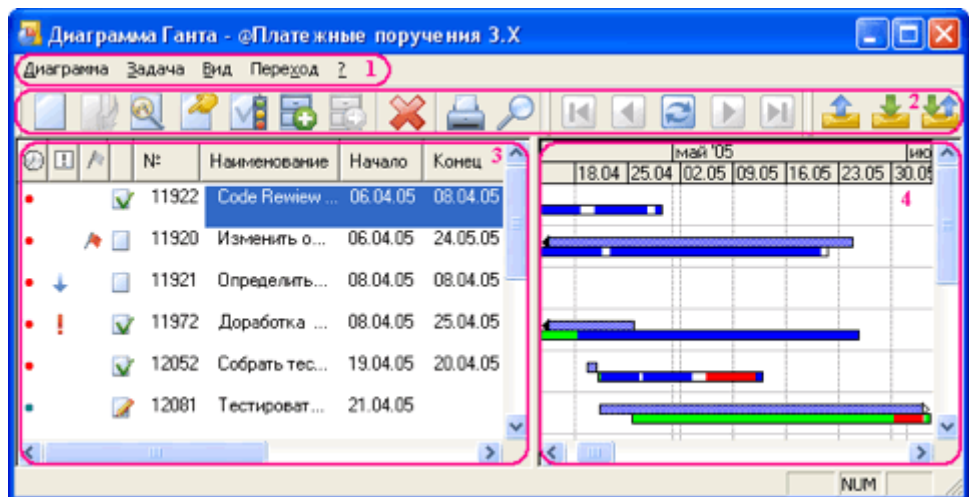


Рис. 9. Диаграмма Ганта в @Управление

Внешний вид диаграмм в программе 1С:Предприятие (рис. 10).

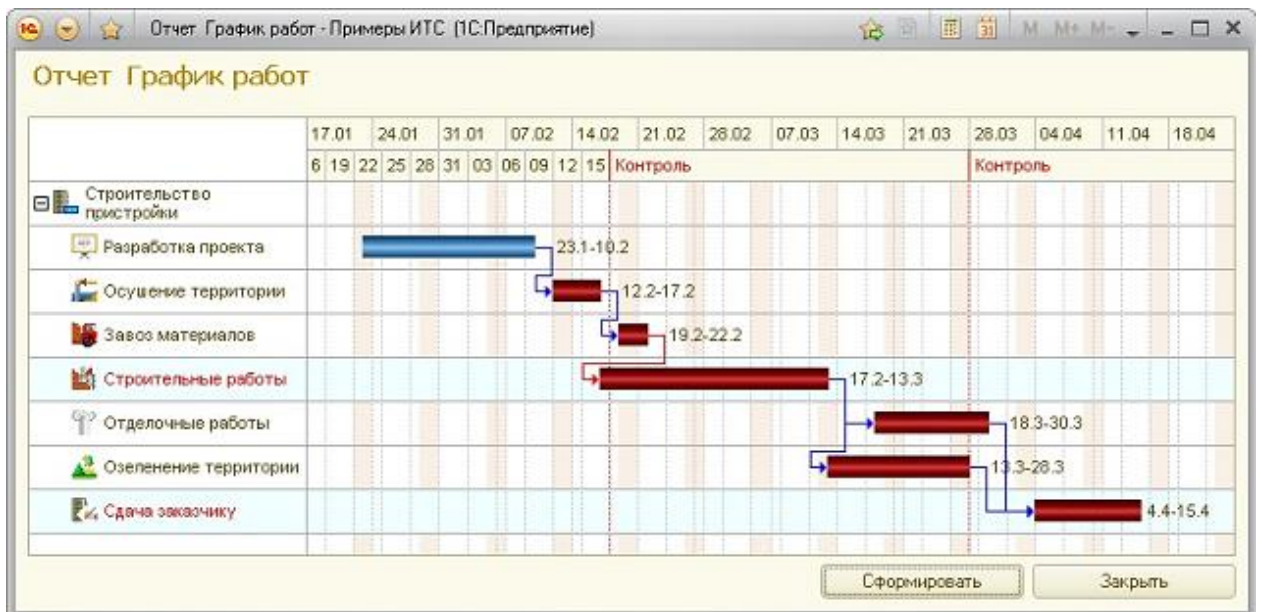


Рис. 10. Диаграмма Ганта в 1С:Предприятие  
Внешний вид диаграмм в программа Адванта (рис. 11).

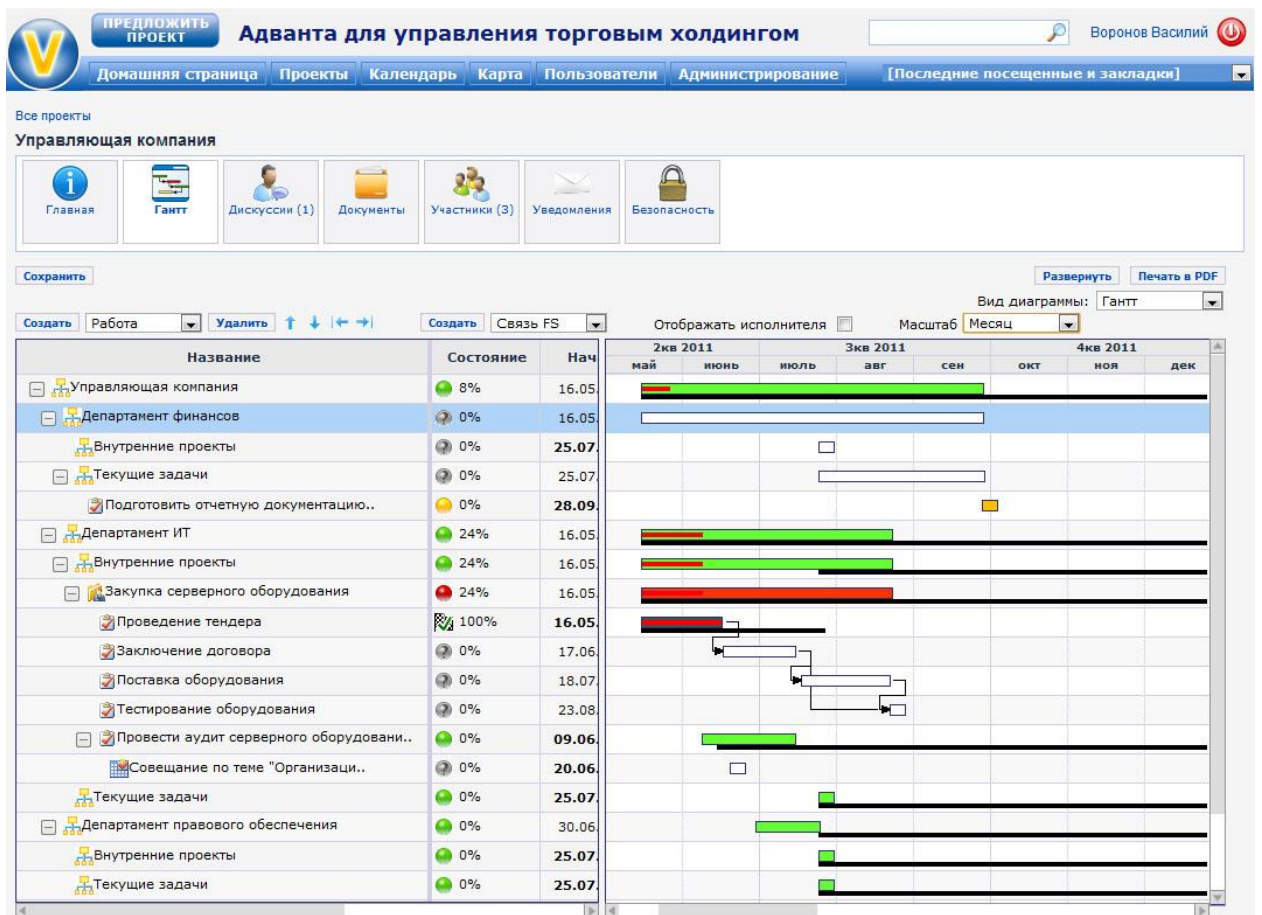


Рис. 11. Диаграмма Ганта в Адванта

## Ход работы

**Задание 1.** Создайте диаграмму Ганта по проекту «Зимняя сессия».

Последовательность действий.

1. Создать и заполнить таблицу с перечислением этапов, датами начала, длительности каждого этапа и конца.
2. Для выделенной таблицы создать диаграмму (тип «Линейчатая с накоплением»).
3. На вкладке «Диапазон данных» выбрать «Ряды в столбцах».
4. На вкладке «Ряд» нажать кнопку «Добавить», установить курсор в поле «Значения» и выделить ячейки с длительностями этапов.
5. Уберите «Легенду».
6. В контекстном меню вертикальной оси с названиями этапов выбрать пункт «Формат оси».
7. На вкладке «Шкала» поставить две галочки: «Обратный порядок категорий» и «Пересечение с осью Y в максимальной категории».
8. Сделайте двойной щелчок мышью по любому из синих столбиков, установите прозрачную заливку и невидимую рамку.
9. Настройте диапазон отображаемых на диаграмме данных. Для этого необходимо узнать реальное содержимое ячеек, с которых начинается и на которых заканчивается временная шкала (первая ячейка в столбце Начало и последняя в столбце Конец). Дело в том, что Excel только отображает в ячейке дату как день-месяц-год, а на самом деле любую дату хранит в ячейке как количество дней, прошедших с 1.1.1900 до текущей даты. Выделите желтую и зеленую ячейки и по очереди попробуйте установить для них «Общий формат» (меню «Формат» – «Ячейки»). Например, получилось 38350 и 38427, соответственно. Прибавим к дате окончания еще три дня – получим 38340. Запомним эти числа.
10. Щелкните правой кнопкой мыши по горизонтальной оси времени и выбрать «Формат оси» и ввести эти числа на вкладку «Шкала»: в минимальное и максимальное значение соответственно.

**Задание 2.** Создайте диаграмму Ганта по следующей таблице, отражающей этапы проекта (рис. 12).



Этап проекта	Начало	Длительность	Задержка	Конец
Организационное собрание	24.04.2011	1	0	24.04.2011
Разработка документации	25.04.2011	11	0	05.05.2011
Общая схема	09.05.2011	3	3	17.05.2011
Разработка модуля 1	18.05.2011	15	0	01.06.2011
Разработка модуля 2	28.05.2011	25	-5	21.06.2011
Разработка модуля 3	26.06.2011	12	4	07.07.2011
Ввод данных	05.07.2011	12	-3	16.07.2011
Анализ данных	17.07.2011	5	0	21.07.2011
Отчет по разработке	22.07.2011	4	0	25.07.2011
Внедрение	24.07.2011	10	-2	02.08.2011
Итоговый отчет	03.08.2011	5	0	07.08.2011
Итоговое собрание	08.08.2011	1	0	08.08.2011

Рис. 12. Образец таблицы планирования

Воспользуйтесь быстрым способом создания диаграмм Ганта.

При помощи условного форматирования можно заставить Excel заливать ячейку любым выбранным цветом, если она по дате попадает между началом и концом этапа. Проще всего для этого использовать логическую функцию И, которая в данном случае проверяет обязательное выполнение обоих условий (5 января позже, чем 4-е и раньше, чем 8-е) (рис. 13).

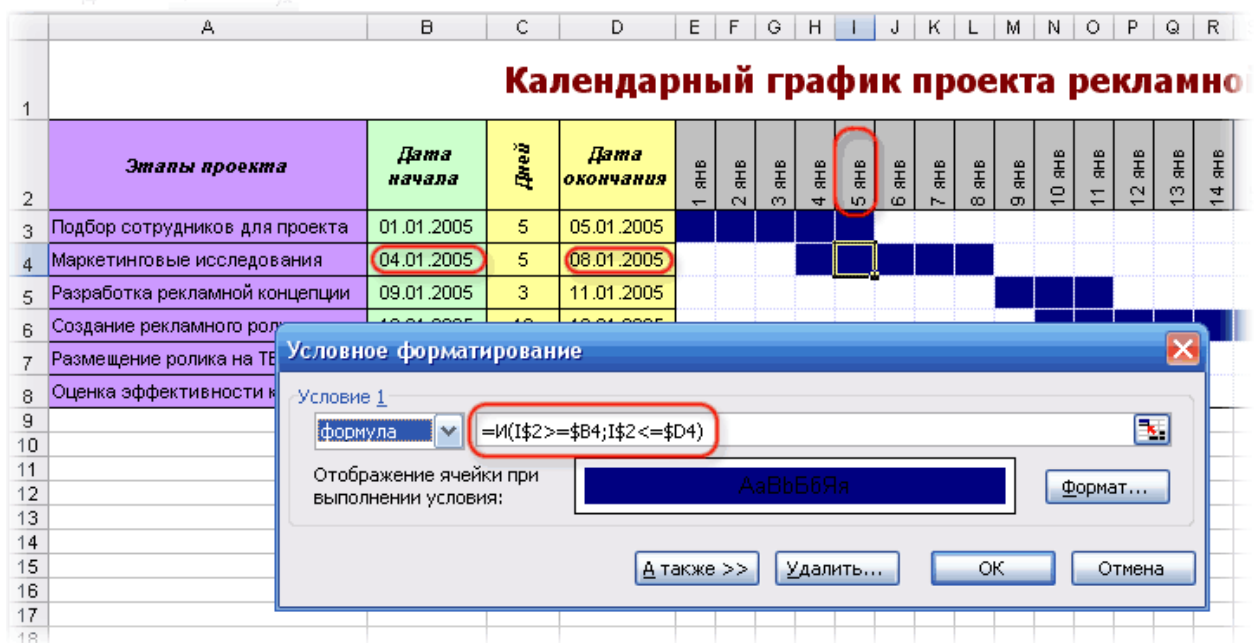


Рис. 13. Пример условного форматирования

**Задание 3 (индивидуально).** Создайте диаграмму Ганта по разрабатываемому вами программному проекту онлайн средствами (Приложение 1).

### Контрольные вопросы

1. Для чего производится временное планирование проекта?

2. Укажите назначение диаграмм Ганта.
3. В чем особенность диаграмм Ганта?
4. Назовите типы связей на диаграмме Ганта, приведите примеры.
5. Какие программными продуктами можно построить диаграммы Ганта?

**По завершении занятия студент должен:**

1. Знать назначение временного планирования проекта.
2. Называть методы разработки временного планирования.
3. Создавать диаграммы Ганта различными способами.
4. Осуществлять временное планирование программного проекта.

## **Лабораторная работа №4**

### **Этап выявления потребностей**

**Цель:** выявить потребности заинтересованных в проекте лиц.

#### **Теоретические вопросы:**

Процесс работы с требованиями к продукту можно разделить на четыре этапа:

1. Определение концепции продукта.
2. Сбор требований.
3. Анализ требований.
4. Проектирование системы.

На этапе определения концепции продукта проводится работа с его инвестором. Цель этапа – выработка единого видения будущего продукта. По окончании этого этапа делается вывод о том, будет ли этот продукт разрабатываться или нет.

На этапе сбора требований ведется основная работа с заказчиком системы и ее будущими пользователями. Цель этапа – точно определить функции продукта и способы его интеграции в существующие процессы.

Качественное выполнение работ на этом этапе гарантирует, что будущий продукт будет соответствовать ожиданиям заказчика. Четкая расстановка приоритетов обеспечивает реализацию наиболее востребованной функциональности и исключение второстепенной/невостребованной функциональности, что экономит бюджет и сроки.

На этапе анализа требований осуществляется структуризация уже собранных ранее требований. Цель этапа – предоставить четкий список не дублируемых требований к системе, которые должны быть выделены из избыточных и частично дублирующихся сценариев и пользовательских историй, полученных на предыдущем этапе.

Правильно сгруппированные требования помогут обойтись минимальным количеством функционала для удовлетворения максимально большего количества целей, а это, в свою очередь, поможет сэкономить бюджет проекта.

На этапе проектирования системы группа разработки принимает проектные решения о том, какую функциональность будет нести продукт,

чтобы удовлетворить пользователей. Результатом служит законченное техническое задание (ТЗ) к продукту: полное описание поведения будущего продукта без неоднозначностей и вопросов.

На основе ТЗ начинается моделирование работы продукта с конечными пользователями и производится его тестирование. Это позволяет увеличить качество продукта и снизить его стоимость, так как стоимость внесения изменений в техническое задание всегда меньше, чем в конечный продукт.

Однако при работе с заказчиками проекта и будущими пользователями программы возникают различные ситуации. Сложившиеся ситуации часто называют «синдромами». Выделяют три основных «синдрома».

1. Синдром «Да... Но!». Эта возникновение противоположных реакций пользователя, когда он впервые видит реализацию системы. С одной стороны, это то, о чем он говорил, с другой – это не то, что он ожидал увидеть.

2. Синдром «неоткрытых руин». Со временем появляется некая закономерность: чем больше выявлено требований, тем больше деталей заказчик стремится добавить в свои требования.

3. Синдром «Говорить на языке пользователя». Во время общения с заказчиком не следует использовать профессионализмы – термины, которые могут быть неизвестны пользователю или неправильно им истолкованы.

Дисциплина управления проектами предлагает различные методы выявления требований, проблем и желаний заказчика:

- интервью и анкеты;
- семинары/совещания (заинтересованные лица собираются для интенсивных, насыщенных дискуссий);
- сценарии приложения (использование визуальных/графических инструментов для демонстрирования поведения системы) для исключения синдрома «Да... Но!»;
- ролевые игры (каждому члену группы назначается определенная роль, обычно роль одного из пользователей);
- метод «мозгового штурма»;
- использование прототипов (как можно раньше предоставить пользователю интерфейс);

- сценарии использования (Use Cases – взаимодействие между пользователем и системой, представленное в виде последовательности шагов);
- анализ существующих документов (извлечение информации из документов Microsoft Word, электронной почты и записей);
- наблюдение и демонстрирование задач (наблюдение за пользователями, выполняющими определенную задачу);
- анализ существующих систем (сбор требований от морально устаревших заменяемых систем или от систем, разработанных в ходе конкуренции).

Рассмотрим особенности метода интервью. Его преимущество – интерактивность, предоставляющая возможность внесения дополнений или доработки вопросов в зависимости от полученных ответов. На сегодняшний день это хороший способ собрать требования по удобству использования системы, надежности, производительности и удобству сопровождения. Обычно заказчики не упоминают эти нефункциональные требования, пока их явно не спросить об этом.

При выборе заинтересованных лиц для интервью следует убедиться, что команда разработчиков понимает, какую именно группу заинтересованных лиц они представляют.

Можно следовать следующим советам.

1. Рекомендуется написать первоначальный список вопросов.
2. Желательно проговорить ответы своими словами, чтобы убедиться, что Вы понимаете смысл.
3. Не следует предлагать ответ на заданный вопрос. (*Какое время реакции системы Вы ожидаете? Три секунды?*).
4. Не следует соединять несколько вопросов в один. (*Необходимо ли Вам печатать ответ, отправлять его по электронной почте и факсу?*). Быть может, пользователю нужна только возможность печати отчета и отправки его по электронной почте, но нет необходимости в факсе.
5. Не следует спрашивать пользователя о деталях реализации. (*Вы предпочитаете list-box или radio-buttons для выбора метода оплаты?*).
6. Не следует использовать слишком длинные и сложные вопросы.

7. Не рекомендуется задавать следующий вопрос, если еще не получен ответ на предыдущий.
8. В ситуации, когда ответ непонятен, стоит задать дополнительные вопросы, даже если их нет в сценарии интервью.
9. Не стоит перебивать пользователей, когда они отклоняются от темы. Надо позволить им высказать свои мысли, на какую бы тему они не размышляли. Если ответ на изначальный вопрос не получен, следует задать его снова.
10. Фиксировать каждое упомянутое пользователем требование, даже если в настоящий момент оно кажется неуместным.
11. Обязательно следует спросить пользователей о дополнительной информации (экранные формы системы).
12. При разговоре с заказчиками не стоит говорить, будет ли их требование выполнено или нет. Это решение можно принять позже.
13. В конце разговора обязательно задайте вопрос для получения дополнительной информации. (*Что еще я должен знать?*).
14. После получения списка требований, следует выяснить у заинтересованного лица приоритет каждого требования.
15. По ходу интервью стоит делать примечания и/или использовать записывающее устройство.
16. Вопросы должны быть контекстно-свободными, т. е. не содержать желаемый ответ.
17. Все требования заносятся в «архив требований», т. е. должны документироваться.

### **Ход работы**

**Задание 1.** Уточните список пользователей и заинтересованных лиц для проекта Автоматизированная информационная система «Университет» (АИС «Университет»). (По желанию студентов можно осуществить реализацию другого программного проекта. Например, «Склад», «Поликлиника», «Сайт организации», «Поиск тура» и др.).

**Задание 2.** Распределите в группе роли согласно списку, полученному в задании 1. Например, Ректор, Проректор, Декан, Заведующий кафедрой, Преподаватель кафедры, Студент, Секретарь деканата и т. д. В соответствии

с ролью изучите возможные должностные обязанности (поиск в сети Интернет), обсудите с преподавателем возможные потребности, проблемы, возникающие с выполнением должностных обязанностей, составьте легенду вашего пользователя.

**Задание 3.** Реализуйте деловую игру. Каждый участник по очереди будет играть роль выбранного им пользователя, остальные – члены команды разработчиков. Методом интервьюирования выявите потребности, проблемы пользователя, подлежащие решению в проекте. Помните о том, что с пользователем необходимо общаться на его языке. Обязательно ведите документирование полученных данных.

**Задание 4.** Согласно своей роли, заполните таблицу описания заинтересованных лиц и пользователей (таблица 9).

Таблица 9 – Характеристика заинтересованного лица

Представитель	Кто в проекте является представителем пользователя? Можно ссылаться на заинтересованных лиц
Описание	Краткое описание типа пользователя
Тип	Уровень знаний пользователя, его техническое образование и степень осведомленности. Например, гуру, случайный пользователь
Ответственность	Список ключевых ответственностей пользователя по отношению к разрабатываемой системе, т.е. фиксирует детали, составляет отчеты, координирует работу и т.д.
Критерий успеха	Как пользователь видит успех? Каким образом компенсируется труд пользователя?
Вовлеченность	Каким образом пользователь может быть вовлечен в проект (рецензирование требований, архитектурных и технических решений, тестирование ПО и т.д.)?
Поставляемые артефакты (документы)	Существуют ли какие-либо выходные артефакты, требуемые пользователю? Если да, то какие (например, отчеты о..., сводка за... и т.д.)?
Комментарии/ Проблемы	Проблемы, мешающие достижению успеха, и любая подобная информация. Можно включать тенденции, которые делают работу пользователя проще или тяжелее.

### Контрольные вопросы

1. Кто такие заинтересованные лица проекта?
2. Как связаны понятия «заинтересованное лицо» и «пользователь»?

3. Какими методами осуществляется выявление потребностей заинтересованных лиц проекта?
4. Каковы преимущества метода интервьюирования?
5. Укажите основные принципы проведения интервьюирования.
6. Всегда ли заказчик проекта имеет представление о реальных потребностях?

**По завершении занятия студент должен:**

1. Знать основные преимущества метода интервьюирования для выявления потребностей заинтересованных лиц проекта.
2. Знать понятие «заинтересованные лица проекта».
4. Иметь представление о принципах проведения интервью с пользователем.
5. Уметь проводить интервью с пользователем.
6. Осуществлять ведение документации: выявленных и зафиксированных потребностей заинтересованных лиц проекта.



## Лабораторная работа №5

### Этап выявления проблемы

**Цель:** выявить возможные проблемы проекта, достигнуть соглашения с заказчиком об определении проблемы.

#### Ход работы

**Задание 1.** Работая в группе разработчиков над АИС «Университет» (или другой, выбранной студентами) с заказчиком (здесь заказчиком выступает преподаватель), выявите проблемы, которые должны быть решены в проекте на основании выявленных потребностей (см. предыдущую лабораторную работу).

**Задание 2.** Представьте, что магазин канцелярии, состоящий из руководителя, продавца, начальника склада и бухгалтера имеет следующую проблему: низкое качество обслуживания клиентов. Предложите возможное решение данной проблемы, заполнив таблицу (таблица 10).

Таблица 10 – Описание проблемы

Проблема:	Описание (в двух-трех предложениях)
Воздействует на:	Указание лиц, на которых оказывает влияние выявленная проблема
Результатом чего является:	Указание причины возникновения проблемы
К чему приводит:	Описание воздействия данной проблемы на заинтересованных лиц и деятельность компании заказчика
Выигрыш от:	Указание предлагаемого командой решения
Успешное решение должно:	Список ключевых преимуществ успешного решения

**Задание 3.** Из полученного в задании 1 списка возможных проблем окончательно определите одну, которая будет решающей для проекта АИС «Университет». Заполните для нее таблицу (таблица 10).

**Задание 4.** Осуществите в сети Интернет поиск основных методов выявления причин проекта. Из предложенных методов выберите один и выделите основные причины существования проекта АИС «Университет» – проблем, стоящих за проблемой.

#### Контрольные вопросы

1. Какими методами возможно выявить проблему заказчика?

2. Всегда ли выявленная проблема совпадает с проблемой, с которой заказчик приходит к разработчику?
3. Для чего необходимо согласование выявленной проблемы с заказчиком?
4. Каким образом прийти к согласованию проблемы с заказчиком, что для этого надо сделать команде разработчиков?
5. Назовите методы выявления причин существования проекта.
6. Что делать, когда выявленная причина проблемы не может быть решена средствами информационных технологий? (Например, проблема – низкая успеваемость в классе, причины – высокая температура в учебных помещениях).

**По завершении занятия студент должен:**

1. Знать особенности выявления проблем заказчика для реализации проекта.
2. Уметь выявлять проблемы заказчика.
3. Уметь выявлять причины существования проекта.
4. Осуществлять анализ проблемы.
5. Осуществлять согласование проблемы.

## **Лабораторная работа №6**

### **Этап анализа проблемы**

**Цель:** выявить заинтересованных лиц проекта.

#### **Теоретические вопросы**

Помимо команды разработчиков проекта и заказчика, существует также множество людей, которые прямо или косвенно получают от решения проблемы прибыль. Их называют заинтересованными лицами проекта (stakeholder). Таким образом, можно выделить:

- инвестора проекта, который оплачивает проект;
- заказчика проекта, который заказывает проект и отвечает за его приемку;
- пользователей проекта, которые будут в конечном счете работать с программой;
- других заинтересованных лиц проекта, которые напрямую или косвенно получают прибыль от решения проекта.

Не всегда инвестор и заказчик являются одним лицом, и не всегда заказчик будет конечным пользователем программы. Следовательно, важно знать всех заинтересованных лиц проекта, их вклад в проект и самое главное – их требования.

В качестве заинтересованных лиц также можно рассматривать [5]:

- любого, участвующего в разработке системы (системные аналитики, дизайнеры интерфейса, программисты, тестеры ПО, менеджеры проекта и др.);
- любого, приносящего свои знания в систему (эксперты предметной области, авторы документов, используемых для сбора требований и др.);
- руководство (руководитель компании-заказчика, руководитель отдела, проектирующего систему);
- лица, непосредственно вовлеченные в управление, настройку и сопровождение системы (например, для сайта это хостинговая компания);
- поставщики регламентов, стандартов.

Помимо интервью рассмотрим еще один метод – анкетирование.

Анкеты наиболее полезны, когда есть возможность задать одни и те же вопросы многим заинтересованным лицам и не надо задавать дополнительные вопросы в процессе беседы. По сравнению с интервью анкетирование обычно влечет за собой меньше расходов. С одной стороны, анкеты структурированы и не интерактивны, с другой – они обеспечивают меньший контроль над результатами.

При составлении вопросов для анкеты следует придерживаться правила: вопросы должны быть понятными и прямолинейными, потому что отсутствует возможность прояснить непонятные моменты или спорные ситуации.

Основная работа состоит не только в разработке анкеты и в проведении анкетирования, но и в обработке результата. Эти процессы можно автоматизировать специальными средствами. Например:

- онлайн-формы Google;
- бесплатный онлайн-сервис <http://webanketa.com/ru/>;
- бесплатный онлайн-сервис <http://www.createsurvey.ru/>;
- и др.

### **Ход работы**

**Задание 1.** После того, как выявлены все заинтересованные лица проекта, а также пользователи, ответьте на следующие вопросы:

1. Кто пользователь будущей автоматизированной системы?
2. Кто заказчик (экономический покупатель программного продукта)?
3. На кого еще окажут влияние результаты работы АИС?
4. Кто будет оценивать и принимать систему после ее установки?
5. Существуют ли другие пользователи, чьи потребности надо учесть (это могут быть как внутренние, так и внешние пользователи)?
6. Кто будет заниматься сопровождением системы?
7. Не забыли ли мы кого-нибудь?

**Задание 2.** После ответов на вопросы при появлении новых пользователей или заинтересованных лиц проекта, которые ранее не были нами учтены, заполните для них таблицу, представляющую их характеристику (таблица 8).

**Задание 3.** Разработайте анкету (с помощью одного из предложенных сервисов) для выбранной вами роли пользователя в проекте АИС «Университет», позволяющую выявить потребности, проблемы, характеристики этого пользователя.

### **Контрольные вопросы**

1. Всех ли пользователей и заинтересованных лиц можно выявить изначально?
2. Какими методами можно выявить неучтенных пользователей и/или заинтересованных лиц проекта?
3. Для чего необходимо знать и иметь характеристику всех пользователей проекта?
4. Каковы особенности метода анкетирования при работе с пользователями проекта? В чем его преимущества, недостатки?

### **По завершении занятия студент должен:**

1. Осуществлять выявление пользователей и заинтересованных лиц проекта, неучтенных на ранних этапах проектирования.
2. Знать основные особенности метода анкетирования пользователей проекта.
3. Уметь составлять анкеты для выявления потребностей пользователя, его характеристик.

## **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).

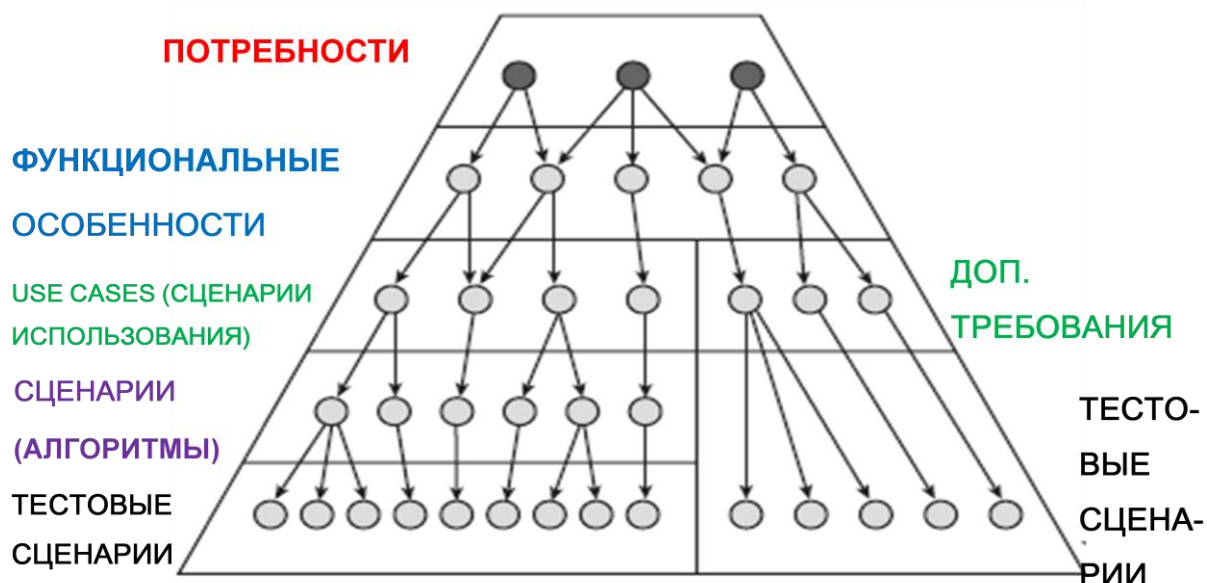


Рис. 14. Пирамида требований

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.

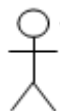


Рис. 15. Нотация эктора на диаграмме

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.

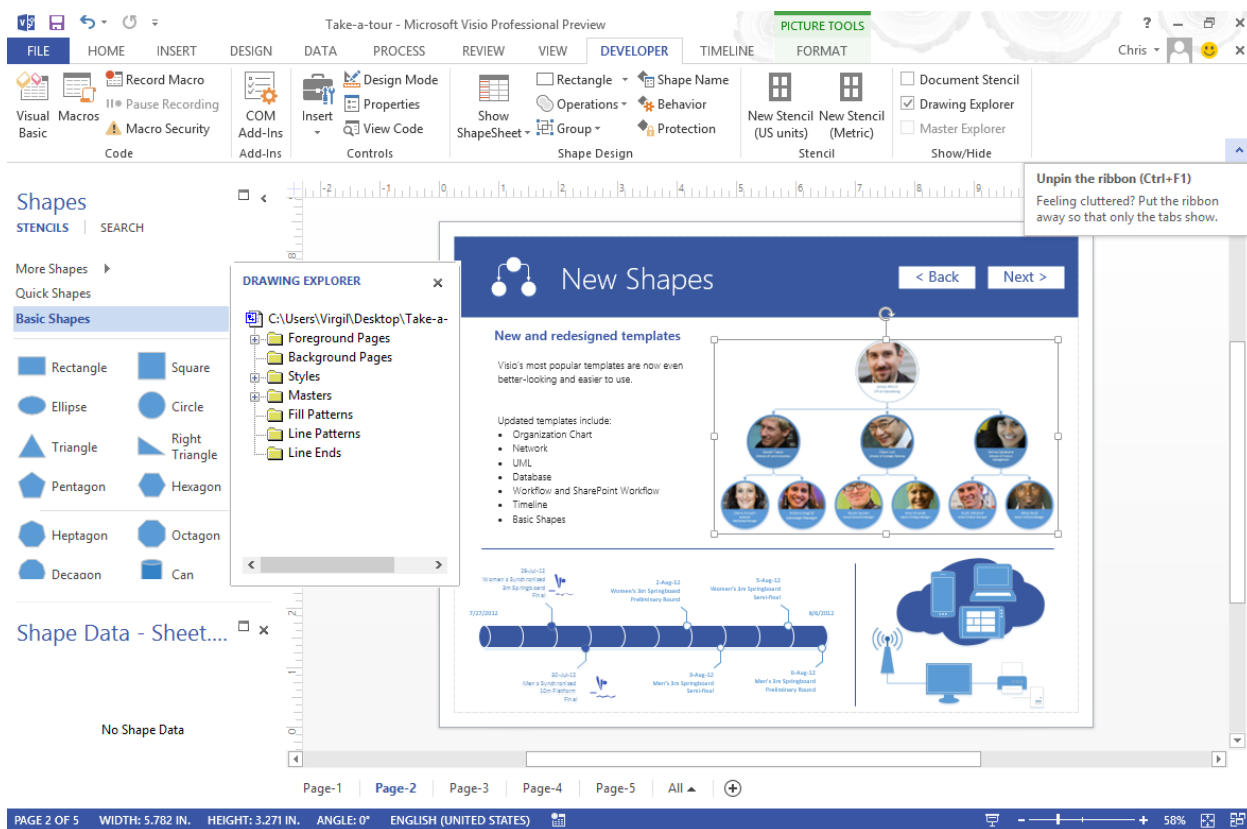


Рис. 16. Векторный редактор Visio

### 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.

## Лабораторная работа № 8

### Сценарии использования

**Цель:** разработка прецедентов следующих уровней – развернутых прецедентов (сценариев использования).

#### Теоретические вопросы

Как уже было сказано ранее, на диаграмме прецедентов между прецедентами и экторами определяются только отношения ассоциации. При этом между элементами диаграммы прецедентов могут существовать различные отношения. Например, взаимодействие экземпляров экторов и вариантов использования.

В языке UML существует несколько стандартных видов отношений между актерами или вариантами использования:

- ассоциации (association relationship);
- расширения (extend relationship);
- обобщения (generalization relationship);
- включения (include relationship).

Обобщение представляет собой отношение между предком и потомком, а стрелка всегда указывает на предка. Кроме того, стрелка направлена в ту сторону, от которой что-то требуется, чьи сервисы используются. Пример отношения обобщения на рис. 17.



Рис. 17. Отношение обобщения между прецедентами

Отношение включения определяет, что внутри базового прецедента содержится поведение другого прецедента. Включаемый прецедент не существует сам по себе, это часть базового. Обозначается в виде пунктирной

линией со стрелкой на конце и подписью «include» (рис. 18). Стрелка направлена в сторону включаемого прецедента.

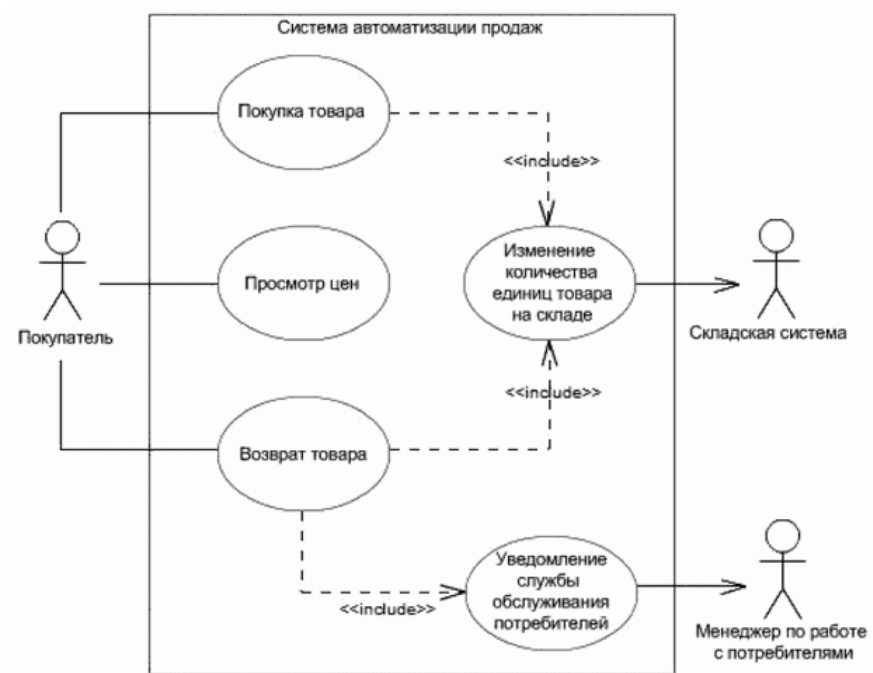


Рис. 18. Пример отношения включения

Такой вид отношения помогает избежать многократного описания одного и того же набора действий, т. к. общее поведение описывается в виде прецедента, включаемого в базовые.

Отношения расширения дополняет прецедент другими прецедентами, выполняемых при некоторых условиях. Это позволяет добавить в исходный прецедент последовательность действий, содержащуюся в другом прецеденте. Обозначается в виде пунктирной линией со стрелкой на конце и подписью «extend» (рис. 19). Стрелка направлена в сторону расширяемого прецедента.



Рис. 19. Пример отношения расширения

При разработке диаграмм прецедентов с отношением расширения можно указать условия возникновения расширенного поведения и точку

расширения прецедента (Extension Point), куда подключаются действия из расширяющих прецедентов (рис. 20).

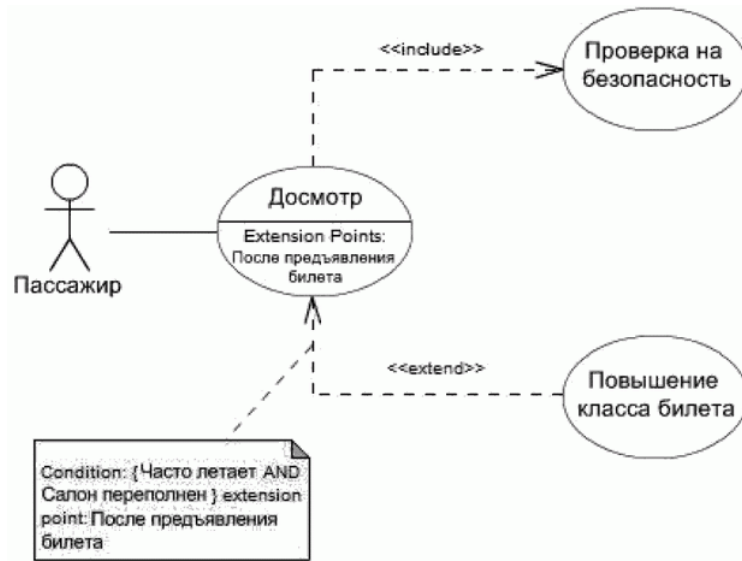


Рис. 20. Обозначения точки расширения

Согласно логике пирамиды требований, после формирования сценариев использования осуществляется разработка сценариев работы. Такие сценарии позволяют перейти от представления о том, «Что сделать», к пониманию того, «Как сделать» (рис. 21).

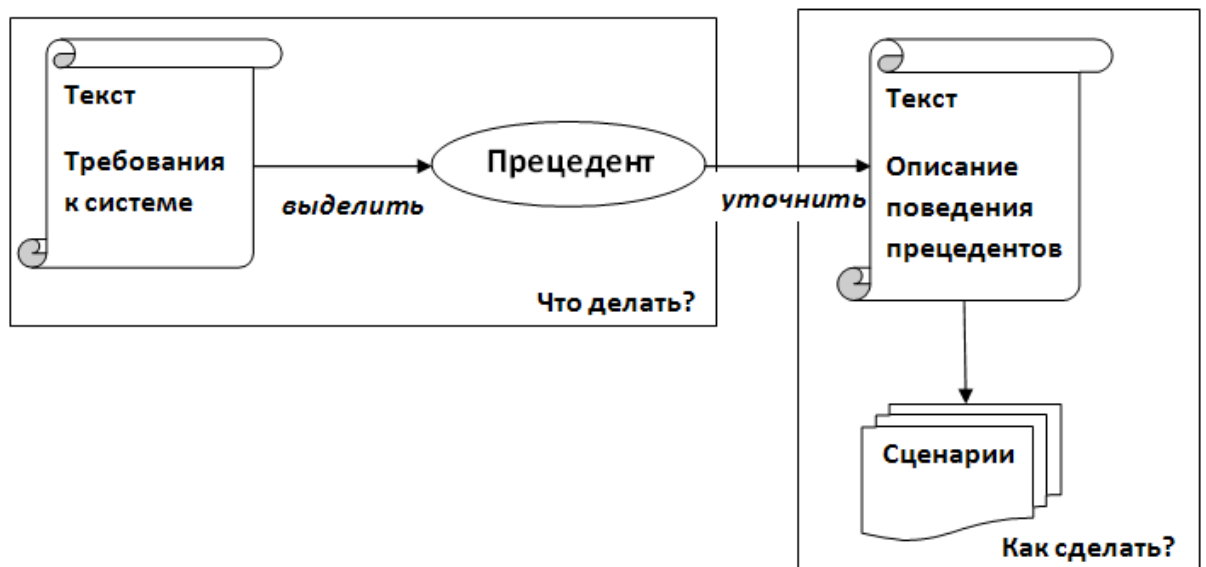


Рис. 21. Переход от сценариев использования к сценариям работы

Сценарий работы представляют собой конкретную последовательность действий, иллюстрирующих поведение системы.

Существует несколько форм сценариев:

1. Текстовая форма.

Пример создания сценария для использования банкомата:

- вставляем карту;
- запрос PIN;
- ввод PIN;
- проверка PIN и т. д.

2. Табличная форма. Пример для использования банкомата в виде таблицы 12.

Таблица 12 – Табличная форма сценария

Действие пользователя	Реакция системы
Вставляем карту	Запрос PIN
Ввод PIN	Проверка PIN

3. Текстовая форма от имени клиента.

Например, как клиент санатория я хочу выбрать номер и забронировать его на определенный срок и ....

### Ход работы

**Задание 1.** Для каждого прецедента первого уровня, полученного на основании предыдущей работы, провести его детализацию в виде диаграммы прецедентов.

**Задание 2.** Произвести документирование каждого из детализированных прецедентов любым из возможных способов. Например, в виде таблицы (таблица 13).

Таблица 13 – Описание развернутого прецедента

Прецедент	Имя прецедента
Исполнители	Перечень исполнителей (внешних агентов), а также те из них, кто инициирует данный прецедент
Цель	Цель прецедента
Краткое описание	Копия содержимого прецедента высокого уровня или некоторая аналогичная обобщенная информация
Тип	1. Главный, второстепенный или дополнительный 2. Идеальный или реальный
Ссылки	Связанные прецеденты и (или) функции системы

**Задание 3.** На основании сформированных диаграмм прецедентов предложить пользователям системы дальнейшие сценарии использования. Их можно оформить в виде таблицы (таблица 14).



Таблица 14 – Сценарии использования

<i>Типичный ход событий</i>	
<b>Действия исполнителя</b>	<b>Отклик системы</b>
Пронумерованные действия исполнителей	Пронумерованные описания откликов системы
<b>Альтернатива</b>	<b>Описание</b>
Альтернатива, которая может возникнуть в строке с номером.	Описание исключения, либо ссылка на соответствующий прецедент

**Задание 4.** Для любого развернутого прецедента создайте сценарий в виде описания действий пользователя.

**Задание 5.** С помощью любых средств нарисуйте PrintScrin для выбранного вами пользователя АИС «Университет» определенного ранее прецедента.

Можно воспользоваться:

1. Средой разработки Visual Studio, Delphi и др.
2. Онлайн сервис для создания внешнего вида мобильных приложений и сайтов <https://moqups.com/home/>.
3. Онлайн сервис для создания диаграмм <https://www.gliffy.com/>.
4. Графические редакторы и др.

### **Контрольные вопросы**

1. Чем отличается прецедент высокого уровня от развернутого прецедента?
2. Какими методами можно документировать развернутые прецеденты?
3. Для чего формируются сценарии использования?
4. Как в дальнейшем могут использоваться эти сценарии?

### **По завершении занятия студент должен:**

1. Знает виды прецедентов: высокого уровня и развернутые.
2. Умеет описывать прецеденты различного уровня.
3. Осуществляет документирование прецедентов различного уровня.
4. Описывает типичный ход событий в сценариях использования, а также предусматривает возможность появления альтернативного хода.

## Лабораторная работа №9

### Ограничения проекта

**Цель:** выявить возможные ограничения проекта АИС «Университет».

#### Теоретические вопросы

Если взглянуть на схему окружения проекта (рис. 22), то можно увидеть, как много всего необходимо учесть при его разработке. В связи с этим на проект и разрабатываемую систему накладываются ограничения.

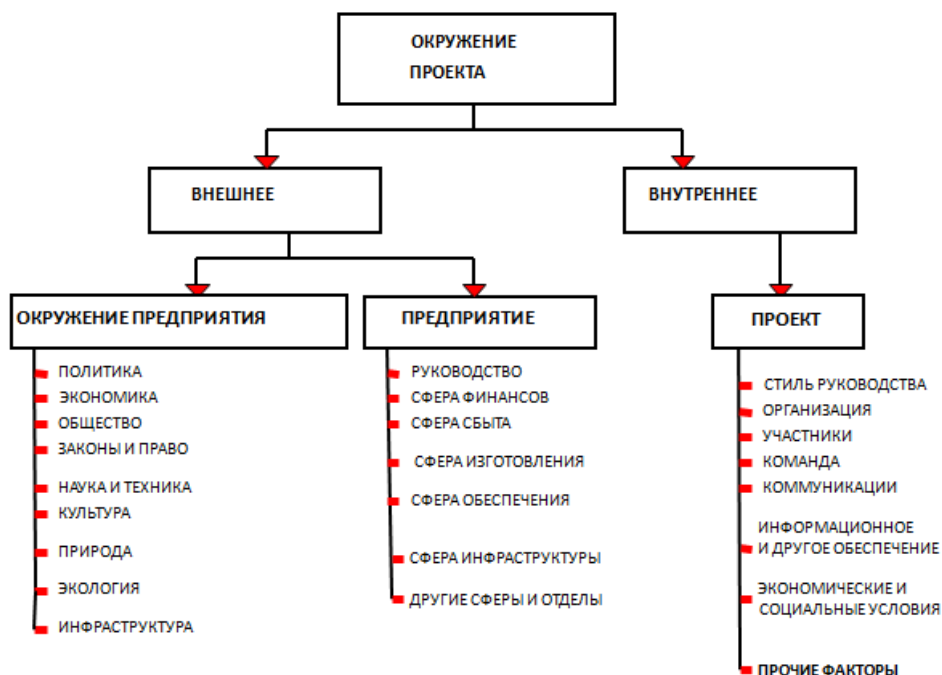


Рис. 22. Окружение проекта

Выделяют следующие ограничения, накладываемые на систему.

1. Экономические. К ним относят бюджет проекта, кроме того, сюда включается лицензия на программное средство разработки, аппаратное обеспечение, заработная плата и т. д.
2. Политические. Данный вид ограничений считается самым сложным. К ним могут относиться различного рода саботажи или особенности внутренней политики компании.
3. Технические. Здесь в качестве примера могут выступать технические мощности парка компьютеров компании, для которой ведется разработка проекта.
4. Системные, т.е. ограничения на стандарты.
5. Эксплуатационные. Это могут быть отключения электроэнергии, ограничения на использование сети Интернет и т. д.

6. График и ресурсы – аспекты, которые касаются графика работы компании, сроков разработки, количества задействованных лиц и т.п.

### **Ход работы**

**Задание 1.** Методом мозгового штурма определите ограничения, накладываемые на АИС «Университет». Результат представьте в виде таблицы (таблица 15).

Таблица 15 – Ограничения проекта АИС «Университет»

<b>Вид ограничения</b>	<b>Ограничение проекта</b>
Экономические	
Политические	
Технические	
Системные	
Эксплуатационные	
График и ресурсы	

**Задание 2 (индивидуально).** Представьте в таблице (таблица 15) ограничения вашего проекта (Приложение).

### **Контрольные вопросы**

1. Что такое ограничение проекта?
2. Перечислите виды ограничений проекта.
3. Приведите примеры ограничений различных видов.

### **По завершении занятия студент должен:**

1. Знать понятие «ограничение проекта».
2. Иметь представление о видах ограничений.
3. Приводить примеры ограничений проекта.

## Лабораторная работа № 10-11

### Техническое задание для проекта АИС «Университет»

**Задание 1.** Изучите значение термина «Техническое задание» с помощью сети Интернет. Дайте определение.

**Задание 2.** Определите документы, регламентирующие составление технического задания на территории РФ.

**Задание 3.** Определите преимущества и недостатки составления технического задания.

**Задание 4.** Скачайте с сайта <http://www.mastertz.ru/> программу «Мастер Технического Задания», изучите с помощью поиска в сети Интернет документацию по техническому заданию (ГОСТ 19, ГОСТ 34).

**Задание 5.** Распределите роли в группе и с помощью программы «Мастер Технических Заданий» создайте документ Техническое задание для проекта АИС «Университет».

1. Из распакованного архива запустите программу `mastertz.exe`. Внешний вид программы представлен на рис. 23.

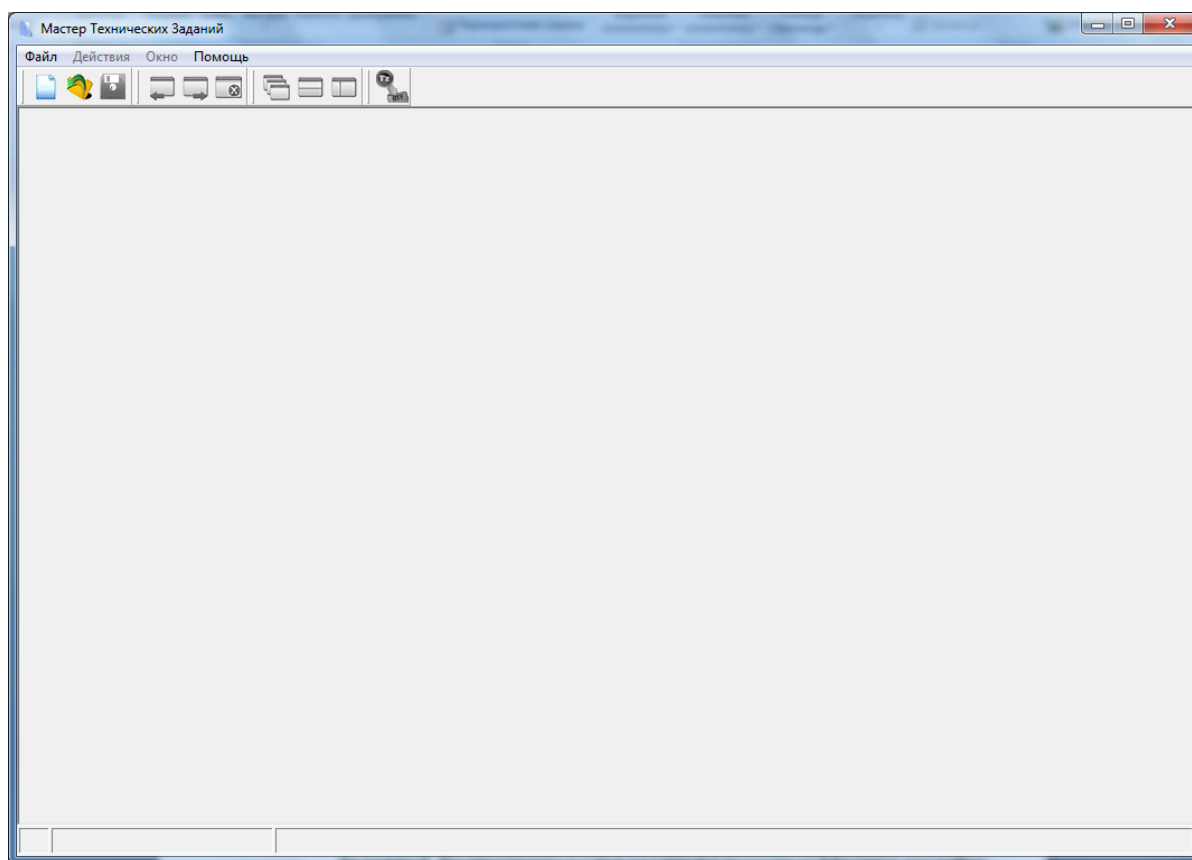


Рис. 23. Внешний вид программы

2. На панели инструментов нажмите кнопку для создания нового ТЗ. Откроется следующая форма (рис. 24).

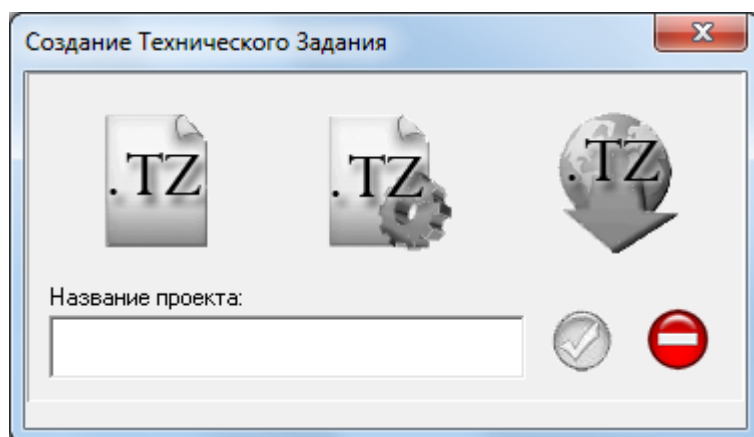


Рис. 24. Окно создания ТЗ

3. Введите название проекта и выберите вид ТЗ (рис. 25).

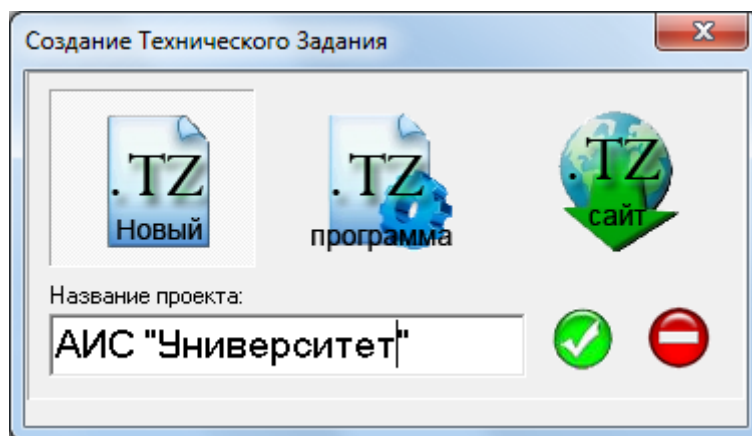


Рис. 25. Выбор ТЗ

4. По нажатию вида ТЗ откроется форма, которую необходимо заполнить (рис. 26).

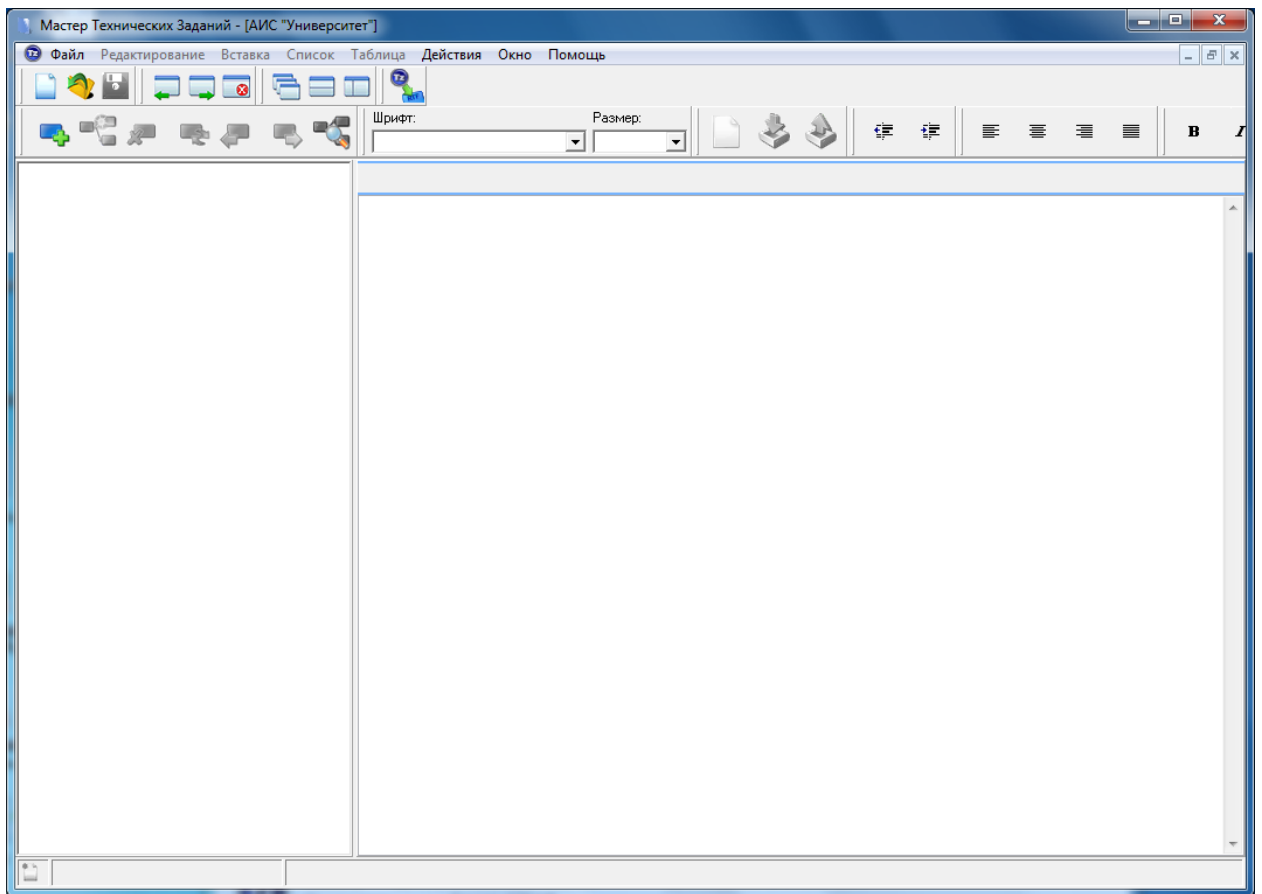


Рис. 26. Форма заполнения ТЗ

5. По завершению ввода данных нажмите кнопку экспорта данных для формирования печатной формы ТЗ (рис. 27).

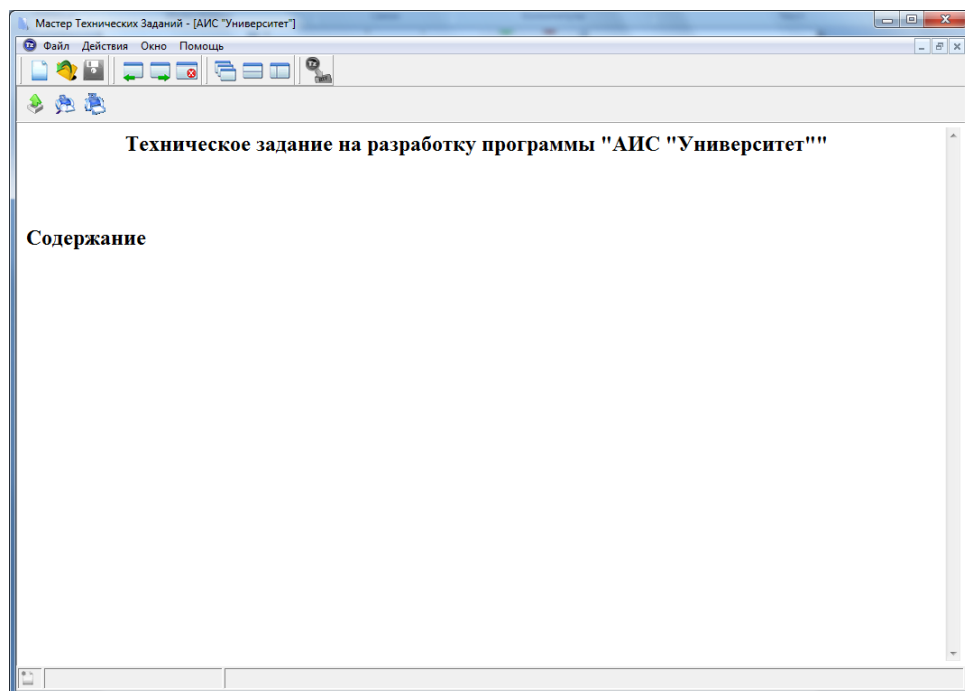


Рис. 27. Печатная форма ТЗ

**Задание 6.** Защитите техническое задание.

**Контрольные вопросы**

1. Что такое техническое задание?
2. На основании каких документов формируется техническое задание для разработки автоматизированной информационной системы?
3. Приведите примеры программных продуктов для формирования технического задания.

**По завершении занятия студент должен:**

1. Знать понятие «техническое задание».
2. Иметь представление о содержании ГОСТ 19, ГОСТ 34.
3. Приводить примеры содержания технического задания информационной системы.

## Лабораторная работа № 12–13

### Реинжиниринг информационных систем

**Цель:** осуществлять реинжиниринг информационных систем.

**Задание 1.** Распределитесь в группы по 3-4 человека. Выберите одну из информационных систем для проведения анализа.

1. 1С:Университет <http://solutions.1c.ru/catalog/university>.
2. АИС «Университет» <http://site.bsu.ru/univer/index.html>.
3. Университетская информационная система НГУ  
<http://www.nsu.ru/exp/university/uis>.
4. Информационная система вашего университета.

По выбранной системе представьте полный анализ (таблица 16).

Таблица 16 – Анализ информационной системы

Разработчик	
Требования к системе	
Стоимость	
Возможности	
Примеры внедрения	
Недостатки	

**Задание 2.** Для анализируемой системы составьте список возможных пользователей. Для каждого пользователя представьте список возможностей в системе.

**Задание 3.** На основании полученного списка возможностей составьте список требований одного пользователя к системе.

**Задание 4.** На основании списка требований к системе создайте диаграмму прецедентов.

**Задание 5.** Выберите один прецедент и детализируйте его: уточните диаграмму прецедентов. Создайте документацию к нему.

**Задание 6.** На основе детализации составьте сценарий работы пользователя, используя Printscreen'ы продукта.

**Задание 7.** Подумайте, какие еще возможности, требования могут появиться у пользователя в ходе работы с системой. Выявились ли в ходе реинжиниринга недостатки системы?

#### Контрольные вопросы

1. Что такое реинжиниринг?



2. С какой целью проводится реинжиниринг?

**По завершении занятия студент должен:**

1. Знать понятие «реинжиниринг».
2. Иметь представление о возможностях методах реинжиниринга при проектировании программных продуктов.

## Лабораторная работа №14-16

### Конвейер проектов

**Цель:** использование методов программной инженерии при разработке курсового проекта.

#### Теоретические вопросы

Наверно, вы уже сталкивались в сети Интернет с таким комиксом (рис. 28). Он хоть и в шуточной форме отражает порой реальный процесс проектирования. Следовательно, задача группы разработчиков прийти к единому пониманию и видению проекта.

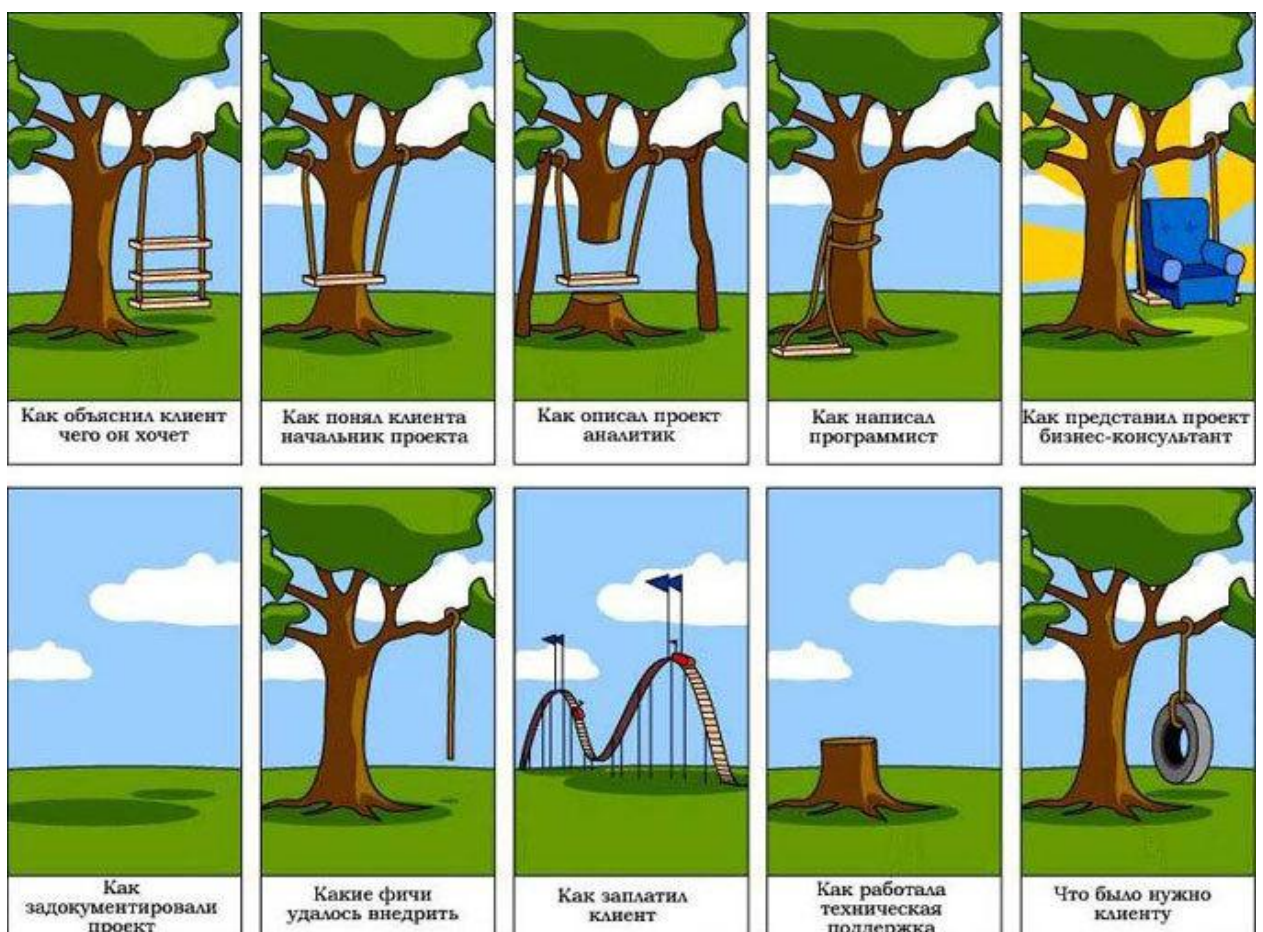


Рис. 28. Действия группы разработчиков проекта

После того как проект создан, его надо подготовить к передаче заказчику. Если это касается курсового проекта, то можно говорить о его презентации. Презентация также понадобится, если хотите представить проект будущим инвесторам, т. е. провести коммерциализацию проекта.

Самым наглядным пониманием того, что сам проект и его презентация – разные вещи, является инсталляция из музея современного искусства (рис. 29).



Рис. 29. Инсталляция

Следовательно, даже самый плохой проект может произвести хорошее впечатление, если его правильно презентовать. Однако на самом деле – презентация не главное (рис. 30).

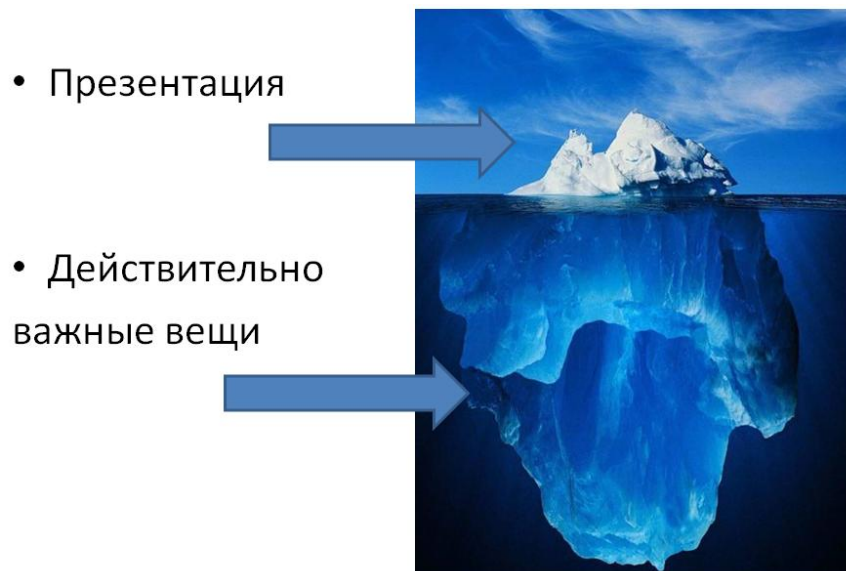


Рис. 30. Представление презентации проекта

В качестве презентации можно представить несколько слайдов, распечатку или тизер проекта (под тизером понимается небольшое предложение о проекте). Однако в самой презентации не всегда указывают модель бизнеса и его детали (например, способы продвижения проекта, его производственный цикл, будущих клиентов, способы их обслуживания, необходимые ресурсы, возможных партнеры), планы работы, бюджет и т. п.

**Задание:** представьте тизер вашего проекта. Например, Instagram – это бесплатное мобильное приложение для обмена, создания, редактирования и распространения фото и видео с элементами социальной сети.

Если вы составляете распечатку особенностей своего проекта, то необходимо уложиться на одну страницу формата А4 (например, рис. 31).

**Резюме проекта**

**Shopfee** — это приложение для смартфонов (iPhone, Android), оно свяжет мир обычных магазинов, кафе и супермаркетов с интерактивным миром интернета и мобильных устройств. Мы хотим адаптировать для российского рынка бизнес-модель недавно запущенных и быстро развивающихся американских проектов: shopkick, checkpoints, weaward, а также французского sclubbucks. Основной вопрос, который задавали себе их создатели:

— Почему никто не прощает людей за посещение магазина?

Эти проекты впервые предоставили возможность магазинам вознаграждать своих посетителей. Это стало возможным благодаря широкому распространению смартфонов. Магазины будут оплачивать foot-traffic — реальных посетителей.

---

**Аудитория Shopfee**

Пользователи приложения получают бонусные баллы:

- за вход в магазин;
- за фотографирование штрих-кодов определенных товаров;
- за перемещения по магазину определенным образом.

Далее он может потратить накопленные бонусные баллы на различные подарочные и скидочные карты, билеты в кино, товары или валюту фейсбука.

Пользователи сфер торговли

**Задача Shopfee**

- Помочь продавцам и покупателям активно взаимодействовать друг с другом посредством iPhone и Android;
- Определить местоположение и действия человека внутри магазина и дать возможность магазину поощрить клиента за эти действия.
- Позволить каждому пользователю почувствовать себя VIP-клиентом. Даже если таких как он несколько тысяч;

**Различные магазины**

Магазин имеет возможность:

- выставлять сумму вознаграждения за чек-ин, тем самым регулируя фут-трафик;
- выбирать товары, на которые необходимо обратить внимание покупателя;
- управлять перемещением покупателя в магазине;
- информировать покупателя об акциях и скидках.

Партнеры: АЗБУКА ВКУСА, Спортмастер, М.Видео, Золотой Ястреб

---

**Бизнес-модель**

- Магазины оплачивают бонусные баллы, которые набирает покупатель за чек-ин и определенные действия внутри самого магазина.
- Пользователь обменивает баллы на конкретные бонусы: подарочные и скидочные карты, билеты в кино, различные товары, etc.
- Люди посещают магазины и выполняют там определенные задания для получения бонусных баллов.

---

**Успехи и опыт американского аналога — Shopkick**

С момента запуска в августе 2010 года shopkick установил свое оборудование в 3000 магазинов и ТЦ. За год работы пользователями приложения было совершено 2 млн. посещений магазинов и около 700 млн. просмотров товаров в приложениях. Shopkick превратил сам процесс покупок в увлекательную игру, в которой можно получить реальные призы.

Партнеры shopkick: торговые центры, магазины и бренды ищут о результатах сотрудничества:

- TARGET** *Инновационный подход Shopkick'a позволил нам с большей точностью дифференцировать и награждать своих покупателей — именно этого они ждали от нас.*
- Crate&Barrel** *Shopkick помог нам расширить маркетинговые возможности благодаря использованию мобильных устройств. Результаты оказались действительно феноменальными.*
- kraft foods** *Мобильные изменяют сам процесс шопинга — добавляют элемент игры и фана. Мы увидели, что у технологии shopkick'a большой потенциал влияния на покупателей.*

---

**Текущее состояние проекта Shopfee**

Уже создан полностью рабочий прототип iPhone-приложения, разработана и протестирована технология точного регистрации местоположения человека с помощью ультразвуковых маяков (GPS для этого не достаточно точен). Мы готовы продемонстрировать Shopfee в действии.

---

**Команда проекта**

	<b>Сергей Баркалов</b> управление IT-проектом Образование: МФТИ, РЭШ Опыт работы: iDeas, ConnectLia, FunSpace, NEXTstore Эл. почта: sbarkalov@gmail.com Тел.: +7 (926) 035-39-62		<b>Павел Ершов</b> дизайн, аэробилити Образование: МФТИ, РЭШ Опыт работы: NETCrackin, iDeas, ConnectLia, FunSpace, NEXTstore Эл. почта: pavel@ershov.org Тел.: +7 (915) 184-53-64		<b>Алексей Лебедев</b> маркетинг, электроника Образование: МИФИ Опыт работы: НИИСи РАН, BrandMail, NEXTstore Эл. почта: lbebedev@gmail.com Тел.: +7 (926) 472-39-23
---	---	---	--	---	--

Рис. 31. Пример информации о проекте

При создании презентации проекта для инвесторов, можно придерживаться следующей структуры:

- титульный слайд (название проекта, контакты команды);
- команда (состав команды, контакты членов команды, можно фото);
- содержание проекта (основные предложения, целевая аудитория, решаемая проблема);
- состояние рынка (аналоги или похожие компании, с указанием объемов рынка, цен на программные продукты и т. п.);
- финансовая модель (необходимая сумма, статьи расходов).

Необходимо уложиться в 10-12 слайдов. Изложение должно быть простым, понятным, кратким, сопровождаться фактами и аргументироваться

цифрами. Не стоит обращать внимание на дизайн, достаточно аккуратного исполнения, своего стиля и наглядности.

## Ход работы

### Часть 1. Самостоятельная работа

**Задание 1.** Формулировка проекта.

Тема курсового проекта	
Название программы	

**Задание 2.** Определите все заинтересованные лица Вашего проекта: кому это нужно (целевая аудитория, заказчик, конечные пользователи).

Заинтересованные лица проекта	
Целевая аудитория	
Заказчик	
Конечные пользователи	
Кто-нибудь еще?	

**Задание 3.** Сроки проекта: как срочно нужно / в какие сроки.

Начало проекта	
Окончание проекта	

**Задание 4.** Идея проекта: что именно изменить / сделать / добавить / улучшить / создать / реализовать необходимо в Вашем проекте.

Идея проекта	

**Задание 5.** Проблема, потребность, решаемая проектом.

Изучите пирамиду потребностей (рис. 33):



Рис. 33. Пирамида потребностей по Маслоу

Определите уровень потребности, удовлетворяемой программным продуктом. Таких потребностей может быть несколько.

Удовлетворяемая потребность

Сформулируйте, какую проблему решает проект. Внимательно изучите список заинтересованных лиц проекта (см. задание 1) и укажите, чью именно проблему решает проект. Проект может решать несколько проблем различных заинтересованных лиц.

Решаемая проблема	
Проблема	Заинтересованное лицо

**Задание 6.** «Примерка» идеи на проблему.

Примерим идею (см. задание 4) на проблему (см. задание 5). Обеспечивает ли формулировка идеи проекта принципиальное решение описанной проблемы? Если нет, то корректируем, «подгоняем» идею.

Откорректированная идея

**Задание 7.** Компромиссы проекта. Заполните матрицу компромиссов.

	Фиксируется	Согласовывается	Принимается
Время			
Ресурсы			
Возможности			

**Задание 8.** Проведите SWOT-анализ курсового проекта.

Метод анализа проекта. Аббревиатура из терминов на английском языке:

- Strength – сильные стороны. Совокупность положительных факторов/преимуществ, положительных условий, имеющихся в вашем распоряжении;
- Weakness – слабые стороны. Отсутствующие в наличии ресурсы, неблагоприятные условия, конкуренты;
- Opportunities – возможности. Наличие ресурсов и/или наличие нескольких кондиций, которые могут быть привлечены (использованы) в целях проекта при определенных условиях с нашей стороны;
- Threats – угрозы/риски. Негативные вероятности, которые могут препятствовать реализации целей проекта и/или существенно изменять ключевые параметры в худшую сторону.

К внутренним факторам относятся сильные и слабые стороны проекта. К внешним факторам – возможности и угрозы.

Сформулируйте все пункты SWOT-анализа и заполните таблицу.

Внутренние факторы	Внешние факторы
Сильные стороны: 1. 2. ...	Возможности: 1. 2. ...
Слабые стороны: 1. 2. ...	Угрозы: 1. 2. ...

Проведите анализ. Соотнесите сильные стороны с угрозами проекта. Выделите ваши выигрышные позиции. Как предотвратить угрозы проекта?

Соотнесите слабые стороны с возможностями. Как перевести слабые стороны в сильные, используя описанные возможности?

**Задание 9.** Анализ рисков курсового проекта.

Заполните для Вашего курсового проекта матрицу рисков Эйзенхауэра: приведите минимум по два риска разного уровня:



Уровень	Категория	Ключевые параметры
1	АА	Срочно Важно
2	АВ	Срочно Не важно
3	ВА	Не срочно Важно
4	ВВ	Не срочно Не важно

### Задание 10. Фазы проекта.

Разбейте курсовой проект на фазы в зависимости от решаемых задач. Например, создание прототипа программы, формирование базы данных, наполнение базы данных, подбор информации для первой главы, оформление текста работы, тестирование программы и т. д. Для каждого этапа определите сроки его начала и окончания, длительность.

Сроки проекта				
№ п/п	Название этапа	Дата начала	Длительность	Дата окончания
1				
2				
...				

По заполненной таблице создайте диаграмму Ганта.

## Часть 2. Работа в группах

### Задание 11. «Критический взгляд».

Подготовьте выступление о проекте для группы заинтересованных лиц. Для этого расскажите группе идею проекта, его миссию, раздайте роли пользователей проекта. Если есть возможность, пригласите руководителя курсовой в качестве заказчика проекта. Помните, разговаривать с заинтересованными лицами нужно только на «их» языке. Обсудите в группе проект. При выявлении несогласованностей, неточностей, недоработок, внесите изменения в первую часть лабораторной.

Длительность обсуждения – 10 минут для каждого проекта.

**Задание 12.** Создание printscreen'ов программы (индивидуальная работа).

Для каждого пользователя создайте примерный внешний вид приложения: учтите цветовую гамму, размеры шрифта, эргономику, размещение кнопок и т. д. Вспомните все неудобства, которые Вы испытываете, работая в других программах. «Примерьте» дизайн программы «на себя». Ответьте на следующие вопросы:

- нравится ли вам разработанный продукт?
- удобен ли интерфейс?
- все ли требования заказчика вы учли?

Если ответы на вопросы исчерпаны, тогда можно презентовать проект.

### **Задание 13. Обсуждение дизайна программы.**

Работайте в группе. Обменяйтесь printscreen'ами, обсудите. Внесите коррективы.

## **Часть 3. Конвейер проектов**

**Задание 14.** Защитите свой проект перед группой экспертов так, чтобы инвесторы захотели вложить в него деньги.

Для защиты создайте 3-х минутную презентацию курсового проекта на листе формата А4. Все требования к презентации представлены выше в теоретических вопросах.

### **Контрольные вопросы**

1. Что такое единое видение проекта?
2. Перечислите способы достижения единого видения проекта?
3. Что такое тизер проекта?
4. Назовите требования к презентации проекта.
5. Каковы особенности проектирования курсовой работы?
6. Что такое пирамида потребностей?
7. В чем особенности SWOT-анализа? Как его проводить?
8. Для чего предназначена матрица рисков Эйзенхауэра? Как ее применять в процессе проектирования?

### **По завершении занятия студент должен:**

1. Иметь представление об едином видении проекта.
2. Перечислять способы достижения единого видения проекта.

3. Знать понятие «тизер проекта».
4. Создавать тизеры проекта.
5. Иметь представление о требованиях к презентации проекта.
6. Перечислять особенности проектирования курсовой работы.
7. Знать понятие «пирамида потребностей».
8. Проводить SWOT-анализ проекта.
9. Иметь представление о матрице рисков Эйзенхауэра, применять ее в процессе проектирования.

**Списки информационных систем для проектирования**

**Вариант 1. Регистратура поликлиники.**

На примере системы «ТМ:Регистратура»:  
<http://1oms.ru/themes/trustmed2/material.asp?folder=2038&matID=2258>.

**Вариант 2. Обработка обращений граждан для службы ЖКХ.**

На примере «ПитерСофт: Управление процессами» (Бизнес-процесс «Обработка обращений граждан»): <http://piter-soft.ru/automation/more/examples/obrabotka-obrascheniy-grazhdan/>.

**Вариант 3. Электронный документооборот небольшой фирмы.**

На примере «ПитерСофт: Управление процессами» в управлении электронным документооборотом: <http://piter-soft.ru/automation/cases/docs/>.

**Вариант 4. Продажа авиабилетов.**

Автоматизированная система продажи билетов:  
[http://www.astt.ru/doc/magazine2/avtomat\\_5.doc](http://www.astt.ru/doc/magazine2/avtomat_5.doc).

**Вариант 5. Туристическая фирма.**

На примере компании Мирида-Тур: <http://corpsite.ru/Project/Mirida-Tour.aspx>

**Вариант 6. Билетная система (Музеи).**

На примере TicketNet-M (музеи): <http://www.infotec.ru/ticketnet/ticketnet-m>.

**Вариант 7. Билетная система (Театр)**

На примере АИС «СУПЕРБИЛЕТ – ТЕАТР»:  
<http://www.superbilet.ru/index.php>.

**Вариант 7. Нарушения ПДД.**

На примере системы обработки данных административных материалов по нарушениям ПДД, зафиксированных посредством комплексов фото-видеофиксации: [http://www.rainbowsoft.ru/fvf\\_avangard](http://www.rainbowsoft.ru/fvf_avangard).

**Вариант 8.** Разработать программный модуль «Учет успеваемости студентов». Программный модуль предназначен для оперативного учета успеваемости студентов в сессию деканом, заместителями декана и сотрудниками деканата. Сведения об успеваемости студентов должны

храниться в течение всего срока их обучения и использоваться при составлении справок о прослушанных курсах и приложений к диплому.

**Вариант 9.** Разработать программный модуль «Личные дела студентов». Программный модуль предназначен для получения сведений о студентах сотрудниками деканата, профкома и отдела кадров. Сведения должны храниться в течение всего срока обучения студентов и использоваться при составлении справок и отчетов.

**Вариант 10.** Разработать программный модуль «Решение комбинаторно-оптимизационных задач». Модуль должен содержать алгоритмы поиска цикла минимальной длины (задача коммивояжера), поиска кратчайшего пути и поиска минимального связывающего дерева.

**Вариант 11.** Разработать программный модуль «Кафедра», содержащий сведения о сотрудниках кафедры (ФИО, должность, ученая степень, дисциплины, нагрузка, общественная работа, совместительство и др.). Модуль предназначен для использования сотрудниками отдела кадров и деканата.

**Вариант 12.** Разработать программный модуль «Лаборатория», содержащий сведения о сотрудниках лаборатории (ФИО, пол, возраст, семейное положение, наличие детей, должность, ученая степень). Модуль предназначен для использования сотрудниками профкома и отдела кадров.

**Вариант 13.** Разработать программный модуль «Химчистка». При записи на обслуживание заполняется заявка, в которой указываются ФИО владельца, описание изделия, вид услуги, дата приема заказа и стоимость услуги. После выполнения работ распечатывается квитанция.

**Вариант 14.** Разработать программный модуль «Учет нарушений правил дорожного движения». Для каждой автомашины (и ее владельца) в базе хранится список нарушений. Для каждого нарушения фиксируется дата, время, вид нарушения и размер штрафа. При оплате всех штрафов машина удаляется из базы.

**Вариант 15.** Разработать программный модуль «Картотека автомагазина», предназначенный для использования работниками агентства. В базе содержатся сведения об автомобилях (марка, объем двигателя, дата выпуска и др.). При поступлении заявки на покупку производится поиск

подходящего варианта. Если такого нет, клиент заносится в клиентскую базу и оповещается, когда вариант появляется.

**Вариант 16.** Разработать программный модуль «Картотека абонентов АТС». Картотека содержит сведения о телефонах и их владельцах. Фиксирует задолженности по оплате (абонентской и повременной). Считается, что повременная оплата местных телефонных разговоров уже введена.

**Вариант 17.** Разработать программный модуль «Автокасса», содержащий сведения о наличии свободных мест на автобусные маршруты. В базе должны содержаться сведения о номере рейса, маршруте, водителе, типе автобуса, дате и времени отправления, а также стоимости билетов. При поступлении заявки на билеты программа производит поиск подходящего рейса.

**Вариант 18.** Разработать программный модуль «Книжный магазин», содержащий сведения о книгах (автор, название, издательство, год издания, цена). Покупатель оформляет заявку на нужные ему книги, если таковых нет, он заносится в базу и оповещается, когда нужные книги поступают в магазин.

**Вариант 19.** Разработать программный модуль «Автостоянка». В программе содержится информация о марке автомобиля, его владельце, дате и времени въезда, стоимости стоянки, скидках, задолженности по оплате и др.

**Вариант 20.** Разработать программный модуль «Кадровое агентство», содержащий сведения о вакансиях и резюме. Программный модуль предназначен как для поиска сотрудника, отвечающего требованиям руководителей фирмы, так и для поиска подходящей работы.

## СПИСОК ЛИТЕРАТУРЫ

1. Архипенков С. Лекции по управлению программными проектами [Электронный ресурс]. – URL: [http://citforum.ru/SE/project/arkhipenkov\\_lectures/](http://citforum.ru/SE/project/arkhipenkov_lectures/)
2. Буч, Г. Введение в UML от создателей языка / Г. Буч, Д. Рамбо. И. Якобсон. 2-е изд.: Пер. с англ. Мухин Н. – М.: ДМК Пресс, 2010. – 496 с.
3. ГОСТ Р ИСО 9000-2008. Системы менеджмента качества. Основные положения и словарь [Электронный ресурс]. – URL: <http://vsegost.com/Catalog/47/47957.shtml>
4. Коуберн, А., Каждому проекту своя методология / А. Коуберн // Humans and Technology Technical Report, TR 99.04, Oct.1999 (русский перевод — К. Максимов, А. Максимова).
5. Мазур, И.И. Управление проектами: Учебное пособие / И.И. Мазур, В.Д. Шапиров // Под общ. ред. И.И. Мазура. – 10-е изд. – М.: Омега-Л, 2013. – 960 с.
6. Официальный сайт Института Управления Проектами. РМВОК [Электронный ресурс]. – URL: <http://www.pmi.org/default.aspx>
7. Рассел, Д. Диаграмма Ганта / Д. Рассел. Изд-во: Книга по Требованию, 2012. – 42 с.
8. Свод знаний по управлению проектами [Электронный ресурс]. – URL: <http://www.pmtoday.ru/project-management/pmbok-pmp/pmbok.html>
9. Статистика успешности IT-проектов [Электронный ресурс]. – URL: <http://foykes.com/statistika-uspeshnosti-it-proektov/>
- 10.Хорошев А.Н. Основы системного проектирования [Электронный ресурс]. – URL: [http://www.cfin.ru/management/controlling/sys\\_project.shtml](http://www.cfin.ru/management/controlling/sys_project.shtml)
- 11.7th annual State of agile development survey, 2013 [Электронный ресурс]. – URL: <http://www.versionone.com/pdf/7th-Annual-State-of-Agile-Development-Survey.pdf>
- 12.Agile Manifesto [Электронный ресурс]. – URL: <http://agilemanifesto.org/iso/ru/>
13. Cockburn, A., Crystal Clear: A Human-Powered Methodology for Small Teams. Addison-Wesley, 2005. 336 p.