**Лабораторная работа № 1**

**«Разработка технического задания на создание системы»**

**1.     Цель работы:**

Описать и проанализировать информационную систему, распределить роли в группе разработчиков. Составить и проанализировать требования к информационной системе, оформить техническое задание на разработку программного обеспечения.

**2.      Методические указания**

Лабораторная работа направлена на ознакомление с процессом описания информационной системы и получение навыков по использованию основных методов анализа ИС, а также на  ознакомление с процессом разработки требований к информационной системе и составления технического задания на разработку программного обеспечения, получение навыков по использованию основных методов формирования и анализа требований.

Требования к результатам выполнения лабораторного практикума:

1. наличие описания информационной системы;
2. проведение анализа осуществимости выполнения проекта;
3. наличие заключения о возможности реализации проекта, содержащего  рекомендации относительно разработки системы, базовые предложения по объёму требуемого бюджета, числу разработчиков, времени и требуемому программному обеспечению;
4. наличие диаграммы идентификации точек зрения и диаграммы иерархии точек зрения;
5. наличие пользовательских требований, четко описывающих будущий функционал системы;
6. наличие системных требований, включающих требования к структуре, программному интерфейсу, технологиям разработки, общие требования к системе (надежность, масштабируемость, распределённость, модульность, безопасность, открытость, удобство пользования и т.д.);
7. наличие составленного технического задания.

**3.     Теоретические сведения**

**Общие сведения о разработке программного обеспечения**

Проблемы управления программными проектами впервые проявились в 60-х - начале 70-х годов, когда провалились многие большие проекты по разработке программных продуктов. Были зафиксированы задержки в создании ПО, оно было ненадежным, затраты на разработку в несколько раз превосходили первоначальные оценки, созданные программные системы часто имели низкие показатели производительности. Причины провалов коренились в тех подходах, которые использовались в управлении проектами. Применяемая методика была основана на опыте управления техническими проектами и оказалась неэффективной при разработке программного обеспечения.

Важно понимать разницу между профессиональной разработкой ПО и любительским программированием. Необходимость управления программными проектами вытекает из того факта, что процесс создания профессионального ПО всегда является субъектом бюджетной политики организации, где оно разрабатывается, и имеет временные ограничения. Работа *руководителя программного проекта* по большому счету заключается в том, чтобы гарантировать выполнение этих бюджетных и временных ограничений с учетом бизнес-целей организации относительно разрабатываемого ПО.

*Руководители проектов*призваны спланировать все этапы разработки программного продукта. Они также должны контролировать ход выполнения работ и соблюдения всех требуемых стандартов. Постоянный контроль за ходом выполнения работ необходим для того, чтобы процесс разработки не выходил за временные и бюджетные ограничения. Хорошее управление не гарантирует успешного завершения проекта, но плохое управление обязательно приведет к его провалу. Это может выразиться в задержке сроков сдачи готового ПО, в превышении сметной стоимости проекта и в несоответствии готового ПО спецификации требований.

Процесс разработки ПО существенно отличается от процессов реализации технических проектов, что порождает определенные сложности в управлении программными проектами:

1. *Программный продукт нематериален.* Программное обеспечение нематериально, его нельзя увидеть или потрогать. Руководитель программного проекта не видит процесс "роста" разрабатываемого ПО. Он может полагаться только на документацию, кото­рая фиксирует процесс разработки программного продукта.
2. *Не существует стандартных процессов разработки ПО.* На сегодняшний день не существует четкой зависимости между процессом создания ПО и типом создаваемого программно­го продукта. Другие технические дисциплины имеют длительную историю, процессы разработки технических изделий многократно опробованы и проверены. Процессы создания большинства технических систем хорошо изучены. Изучением же процессов создания ПО специалисты занимаются только последнее время. Поэтому пока нельзя точно предсказать, на каком этапе процесса разработки ПО могут возникнуть проблемы, угрожающие всему программному проекту.
3. *Большие программные проекты - это часто "одноразовые" проекты.* Большие программные проекты, как правило, значительно отличаются от проектов, реализованных ранее. Поэтому, чтобы уменьшить неопределенность в планировании проекта, ру­ководители проектов должны обладать очень большим практическим опытом. Но постоянные технологические изменения в компьютерной технике и коммуникационном оборудовании обесценивают предыдущий опыт. Знания и навыки, накоп­ленные опытом, могут не востребоваться в новом проекте.

Перечисленные отличия могут привести к тому, что реализация проекта выйдет из вре­менного графика или превысит бюджетные ассигнования. Программные системы зачас­тую оказываются новинками как в "идеологическом", так и в техническом плане. Поэтому, предвидя возможные проблемы в реализации программного проекта, следует всегда помнить, что многим из них свойственно выходить за рамки временных и бюджетных ограничений.

**Процесс управления разработкой программного обеспечения**

Невозможно описать и стандартизировать все работы, выполняемые в проекте по созданию ПО. Эти работы весьма существенно зависят от организации, где выполняется разработка ПО, и от типа создаваемого программного продукта. Но всегда можно выделить следующие:

* Написание предложений по созданию ПО.
* Планирование и составление графика работ по созданию ПО.
* Оценивание стоимости проекта.
* Подбор персонала.
* Контроль за ходом выполнения работ.
* Написание отчетов и представлений.

Первая стадия программного проекта может состоять из написания предложений по реализации этого проекта. Предложения должны содержать описание целей проектов и способов их достижения. Они также обычно включают в себя оценки финансовых и вре­менных затрат на выполнение проекта. При необходимости здесь могут приводиться обоснования для передачи проекта на выполнение сторонней организации или команде разработчиков.

*Написание предложений* — очень ответственная работа, так как для многих организа­ций вопрос о том, будет ли проект выполняться самой организацией или разрабатываться по контракту сторонней компанией, является критическим. Не существует каких-либо рекомендаций по написанию предложений, многое здесь зависит от опыт.

На этапе *планирования проекта* определяются процессы, этапы и полученные на каждом из них результаты, которые должны привести к выполнению проекта. Реализа­ция этого плана приведет к достижению целей проекта. Определение стоимости проек­та напрямую связано с его планированием, поскольку здесь оцениваются ресурсы, тре­бующиеся для выполнения плана.

*Контроль за ходом выполнения работ (мониторинг проекта)* — это непрерывный процесс, продолжающийся в течение всего срока реализации проекта. Руководитель дол­жен постоянно отслеживать ход реализации проекта и сравнивать фактические и пла­новые показатели выполнения работ с их стоимостью. Хотя многие организации имеют механизмы формального мониторинга работ, опытный руководитель может составить яс­ную картину о стадии развитии проекта просто путем неформального общения с разра­ботчиками.

Неформальный мониторинг часто помогает обнаружить потенциальные проблемы, которые в явном виде могут обнаружиться позднее. Например, ежедневное обсуждение хода выполнения работ может выявить отдельные недоработки в создаваемом программ­ном продукте. Вместо ожидания отчетов, в которых будет отражен факт "пробуксовки" графика работ, можно обсудить со специалистами намечающиеся программист­ские проблемы и не допустить срыва графика работ.

В течение реализации проекта обычно происходит несколько формальных контроль­ных проверок хода выполнения работ по созданию ПО. Такие проверки должны дать об­щую картину хода реализации проекта в целом и показать, насколько уже разработанная часть ПО соответствует целям проекта.

Время выполнения больших программных проектов может занимать несколько лет. В те­чение этого времени цели и намерения организации, заказавшей программный проект, могут существенно измениться. Может оказаться, что разрабатываемый программный продукт стал уже ненужным либо исходные требования к создаваемому ПО просто устарели и их необходи­мо кардинально менять. В такой ситуации руководство организации-разработчика может при­нять решение о прекращении разработки ПО или об изменении проекта в целом с тем, чтобы учесть изменившиеся цели и намерения организации-заказчика.

Руководители проектов обычно обязаны сами *подбирать исполнителей* для своих проектов. В идеальном случае профессиональный уровень исполнителей должен со­ответствовать той работе, которую они будут выполнять в ходе реализации проекта. Однако во многих случаях руководители должны полагаться на команду разработчиков, которая далека от идеальной. Такая ситуация может быть вызвана следующими причинами:

1. Бюджет проекта не позволяет привлечь высококвалифицированный персонал. В таком случае за меньшую плату привлекаются менее квалифицированные специалисты.
2. Бывают ситуации, когда невозможно найти специалистов необходимой квалифика­ции как в самой организации-разработчике, так и вне ее. Например, в организации "лучшие люди" могут быть уже заняты в других проектах.
3. Организация хочет повысить профессиональный уровень своих работников. В этом случае она может привлечь к участию в проекте неопытных или недостаточно квалифицированных работников, чтобы они приобрели необходимый опыт и по­учились у более опытных специалистов.

Таким образом, почти всегда подбор специалистов для выполнения проекта имеет оп­ределенные ограничения и не является свободным. Вместе с тем необходимо, чтобы хотя бы несколько членов группы разработчиков имели квалификацию и опыт, достаточные для работы над данным проектом. В противном случае невозможно избежать ошибок в разработке ПО.

Руководитель проекта обычно обязан посылать *отчеты* о ходе его выполнения как заказ­чику, так и подрядным организациям. Это должны быть краткие документы, основанные на информации, извлекаемой из подробных' отчетов о проекте. В этих отчетах должна быть та информация, которая позволяет четко оценить степень готовности создаваемого программного продукта.

**В рамках курса «Технология разработки программного обеспечения» выделены следующие роли в группе по разработке ПО:**

* **Руководитель – общее руководство проектом, написание документации, общение с заказчиком ПО**
* **Системный аналитик – разработка требований (составление технического задания, проекта программного обеспечения)**
* **Тестер – составление плана тестирования и аттестации готового ПО (продукта), составление сценария тестирования, базовый пример, проведение мероприятий по плану тестирования**
* **Разработчик – моделирование компонент программного обеспечения, кодирование**

**Планирование проекта разработки программного обеспечения**

Эффективное управление программным проектом напрямую зависит от правильного планирования работ, необходимых для его выполнения. План помогает руководителю предви­деть проблемы, которые могут возникнуть на каких-либо этапах создания ПО, и разработать превентивные меры для их предупреждения или решения. План, разработанный на началь­ном этапе проекта, рассматривается всеми его участниками как руководящий документ, вы­полнение которого должно привести к успешному завершению проекта. Этот первоначаль­ный план должен максимально подробно описывать все этапы реализации проекта.

Процесс планирования начинается, исходя из описания системы, с определения проектных ограничений (временные ограничения, возможности наличного персонала, бюджетные ограничения и т.д.). Эти ограничения должны определяться параллельно с оцениванием проектных параметров, таких как структура и размер проекта, а также распределением функций среди исполнителей. Затем определяются этапы разработки и то, какие результаты документация, прототипы, подсистемы или версии программного продукта) должны быть получены по окончании этих этапов. Далее начинается циклическая часть планирования. Сначала разрабатывается график работ по выполнению проекта или дается разрешение на продолжение использования ранее созданного графика. После этого проводится контроль выполнения работ и отмечаются расхождения между реальным и плановым ходом работ.

Далее, по мере поступления новой информации о ходе выполнения проекта, возможен пересмотр первоначальных оценок параметров проекта. Это, в свою очередь, может привести к изменению графика работ. Если в результате этих изменений нарушаются сроки завершения проекта, должны быть пересмотрены (и согласованы с заказчиком ПО) про­ектные ограничения.

Конечно, большинство руководителей проектов не думают, что реализация их проектов пройдет гладко, без всяких проблем. Желательно описать возможные проблемы еще до того, как они проявят себя в ходе выполнения проекта. Поэтому лучше составлять "пессимистические" графики работ, чем "оптимистические". Но, конечно, невозможно построить план, учитывающий все, в том числе случайные, проблемы и задержки выполнения проекта, поэтому и возникает необходимость периодического пересмотра проектных ограничений и этапов создания программного продукта.

План проекта должен четко показать ресурсы, необходимые для реализации проекта, разделение работ на этапы и временной график выполнения этих этапов. В некоторых ор­ганизациях план проекта составляется как единый документ, содержащий все виды планов, описанных выше. В других случаях план проекта описывает только технологический процесс создания ПО. В таком плане обязательно присутствуют ссылки на планы других видов, но они разрабатываются отдельно от плана проекта.

Детализация планов проектов очень разнится в зависимости от типа разрабаты­ваемого программного продукта и организации-разработчика. Но в любом случае боль­шинство планов содержат следующие разделы.

1. *Введение*. Краткое описание целей проекта и проектных ограничений (бюджетных, временных и т.д.), которые важны для управления проектом.
2. *Организация выполнения проекта*. Описание способа подбора команды разработчи­ков и распределение обязанностей между членами команды.
3. *Анализ рисков*. Описание возможных проектных рисков, вероятности их проявле­ния и стратегий, направленных на их уменьшение.
4. *Аппаратные и программные ресурсы, необходимые для реализации проекта*. Перечень ап­паратных средств и программного обеспечения, необходимого для разработки программного продукта. Если аппаратные средства требуется закупать, приводится их стоимость совместно с графиком закупки и поставки.
5. *Разбиение работ на этапы*. Процесс реализации проекта разбивается на отдельные процессы, определяются этапы выполнения проекта, приводится описание резуль­татов ("выходов") каждого этапа и контрольные отметки.
6. *График работ*. В этом графике отображаются зависимости между отдельными про­цессами (этапами) разработки ПО, оценки времени их выполнения и распределе­ние членов команды разработчиков по отдельным этапам.
7. *Механизмы мониторинга и контроля за ходом выполнения проекта*. Описываются пре­доставляемые руководителем отчеты о ходе выполнения работ, сроки их предостав­ления, а также механизмы мониторинга всего проекта.

План должен регулярно пересматриваться в процессе реализации проекта. Одни части плана, например график работ, изменяются часто, другие более стабильны. Для внесения изменений в план требуется специальная организация документопотока, позволяющая от­слеживать эти изменения.

**Общие сведения о требованиях к информационным системам**

Проблемы, которые приходится решать специалистам в процессе создания программного обеспечения, очень сложны. Природа этих проблем не всегда ясна, особенно если разрабатываемая программная система инновационная. В частности, трудно чётко описать те действия, которые должна выполнять система. Описание функциональных возможностей и ограничений, накладываемых на систему, называется требованиями к этой системе, а сам процесс формирования, анализа, документирования и проверки этих функциональных возможностей и ограничений – разработкой требований.

Требования подразделяются на пользовательские и системные. Пользовательские требования – это описание на естественном языке (плюс поясняющие диаграммы) функций, выполняемых системой, и ограничений, накладываемых на неё. Системные требования – это описание особенностей системы (архитектура системы, требования к параметрам оборудования и т.д.), необходимых для эффективной реализации требований пользователя.

**Первые шаги по разработке требований к информационным системам - анализ осуществимости**

Разработка требований — это процесс, включающий мероприятия, необходимые для созда­ния и утверждения документа, содержащего спецификацию системных требований. Для новых программных систем процесс разработки требований должен начинаться с анализа осуществимости. Началом такого анализа является общее описание системы и ее назначения, а результатом анализа — отчет, в котором должна быть четкая рекомендация, продолжать или нет процесс разработки требований проектируемой системы. Другими словами, анализ осуществимости должен осветить следующие вопросы.

1. Отвечает ли система общим и бизнес-целям организации-заказчика и организации-разработчика?
2. Можно ли реализовать систему, используя существующие на данный момент техно­логии и не выходя за пределы заданной стоимости?
3. Можно ли объединить систему с другими системами, которые уже эксплуатируются?

Критическим является вопрос, будет ли система соответствовать целям организации. Если сис­тема не соответствует этим целям, она не представляет никакой ценности для организации. В то же время многие организации разрабатывают системы, не соответствующие их целям, либо не со­всем ясно понимая эти цели, либо под влиянием политических или общественных факторов.

Выполнение анализа осуществимости включает сбор и анализ информации о будущей системе и написание соответствующего отчета. Сначала следует определить, какая именно информация необходима, чтобы ответить на поставленные выше вопросы. Например, эту информацию можно получить, ответив на следующее:

1. Что произойдет с организацией, если система не будет введена в эксплуатацию?
2. Какие текущие проблемы существуют в организации и как новая система поможет их решить?
3. Каким образом система будет способствовать целям бизнеса?
4. Требует ли разработка системы технологии, которая до этого не использовалась в организации?

Далее необходимо определить источники информации. Это могут быть менеджеры отделов, где система будет использоваться, разработчики программного обеспечения, знакомые с типом будущей системы, техноло­ги, конечные пользователи и т.д.

После обработки собранной информации готовится отчет по анализу осуществимости создания системы. В нем должны быть даны рекомендации относительно продолжения разработки системы. Могут быть предложены изменения бюджета и графика работ по созданию системы или предъявлены более высокие требования к системе.

**Общие сведения о требованиях к информационным системам**

Проблемы, которые приходится решать специалистам в процессе создания программного обеспечения, очень сложны. Природа этих проблем не всегда ясна, особенно если разрабатываемая программная система инновационная. В частности, трудно чётко описать те действия, которые должна выполнять система. Описание функциональных возможностей и ограничений, накладываемых на систему, называется требованиями к этой системе, а сам процесс формирования, анализа, документирования и проверки этих функциональных возможностей и ограничений – разработкой требований.

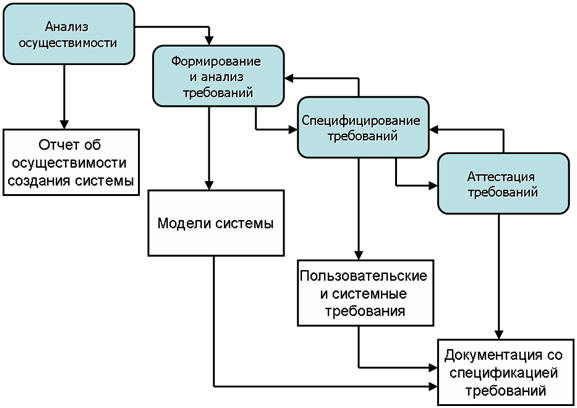
Требования подразделяются на пользовательские и системные. Пользовательские требования – это описание на естественном языке (плюс поясняющие диаграммы) функций, выполняемых системой, и ограничений, накладываемых на неё. Системные требования – это описание особенностей системы (архитектура системы, требования к параметрам оборудования и т.д.), необходимых для эффективной реализации требований пользователя.

**Разработка требований**

Разработка требований — это процесс, включающий мероприятия, необходимые для созда­ния и утверждения документа, содержащего спецификацию системных требований. Различают четыре основных этапа процесса разработки требований:

1. анализ технической осуществимости создания системы,
2. формирование и анализ требований,
3. специфицирование требований и создание соответствующей документации,
4. аттестация этих требований.

На рис. 1 показаны взаимосвязи между этими этапами и результаты, сопровождающие каждый этап процесса разработки системных требований.



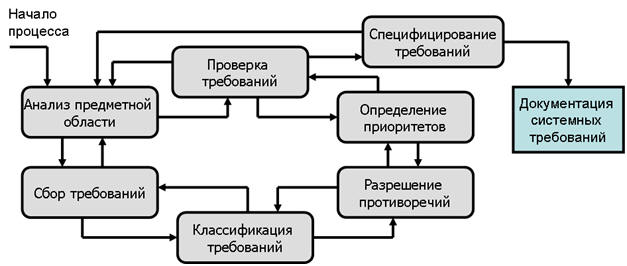
*Рис. 1. Процесс разработки требований*

Но поскольку в процессе разработки системы в силу разнообразных причин требова­ния могут меняться, управление требованиями, т.е. процесс управления изменениями системных требований, является необходимой составной частью деятельности по их раз­работке.

**Формирование и анализ требований**

Следующим этапом процесса разработки требований является формирование (определение) и анализ требований.

Обобщенная модель процесса формирования и анализа требований показана на рис. 2. Каждая организация использует собственный вариант этой модели, зависящий от “местных факторов”: опыта работы коллектива разработчиков, типа разрабатываемой системы, используемых стандартов и т.д.



##### ***Рис. 2. Процесс формирования и анализа требований***

Процесс формирования и анализа требований проходит через ряд этапов.

1. *Анализ предметной области.*Аналитики должны изучить предметную область, где бу­дет эксплуатироваться система.
2. *Сбор требований.*Это процесс взаимодействия с лицами, формирующими требования. Во время этого процесса продолжается анализ предметной области.
3. *Классификация требований.*На этом этапе бесформенный набор требований преобразуется в логически связанные группы требований.
4. *Разрешение противоречий.*Без сомнения, требования многочисленных лиц, занятых в процессе формирования требований, будут противоречивыми. На этом этапе оп­ределяются и разрешаются противоречия различного рода.
5. *Назначение приоритетов.*В любом наборе требований одни из них будут более важ­ны, чем другие. На этом этапе совместно с лицами, формирующими требования, определяются наиболее важные требования.
6. *Проверка требований.*На этом этапе определяется их полнота, последовательность и непротиворечивость.

Процесс формирования и анализа требований циклический, с обратной связью от одного этапа к другому. Цикл начинается с анализа предметной облас­ти и заканчивается проверкой требований. Понимание требований предметной области увеличивается в каждом цикле процесса формирования требований.

Рассмотрим три основных подхода к формированию требований: метод, основанный на множестве опорных точек зрения, сценарии и этнографический метод.

**Опорные точки зрения**

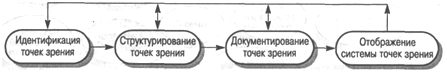
Подход с использованием различных *опорных*точек зрения к разработке требова­ний признает различные (опорные) точки зрения на проблему и использует их в качестве основы построения и организации как процесса формирования требований, так и не­посредственно самих требований.

Различные методы предлагают разные трактовки выражения "точка зрения". Точки зрения можно трактовать следующим образом.

1. *Как источник информации о системных данных.*В этом случае на основе опорных то­чек зрения строится модель создания и использования данных в системе. В процес­се формирования требований отбираются все такие точки зрения ( и на их основе определяются данные), которые будут созданы или использованы при работе систе­мы, а также способы обработки этих данных.
2. *Как структура представлений.*В этом случае точки зрения рассматриваются как особая часть модели системы. Например, на основе различных точек зрения могут разрабатываться модели "сущность-связь", модели конечного авто­мата и т.д.
3. *Как получатели системных сервисов.*В этом случае точки зрения являются внешними (относительно системы) получателями системных сервисов. Точки зре­ния помогают определить данные, необходимые для выполнения системных сер­висов или их управления.

Наиболее эффективным подходом к анализу таких систем является использование внешних опорных точек зрения. На основе этого подхода разработан метод VORD (Viewpoint-Oriented Requirements Definition — определение требований на основе точек зрения) для формирования и анали­за требований. Основные этапы метода VORD показаны на рис. 3.:

1. Идентификация точек зрения, получающих системные сервисы, и идентификация сервисов, соответствующих каждой точке зрения.
2. Структурирование точек зрения — создание иерархии сгруппированных точек зре­ния. Общесистемные сервисы предоставляются более высоким уровням иерархии и наследуются точками зрения низшего уровня.
3. Документирование опорных точек зрения, которое заключается в точном описании идентифицированных точек зрения и сервисов.
4. Отображение системы точек зрения, которая показывает системные объекты, оп­ределенные на основе информации, заключенной в опорных точках зрения.



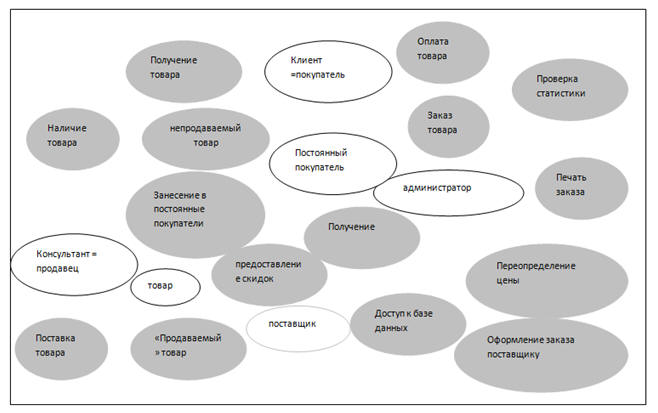
*Рис. 3. Метод VORD*

*Пример.*Рассмотрим использование метода VORD на первых трех шагах анализа требований для системы системы поддержки заказа и учета товаров в бакалейной лавке. В бакалейной лавке для каждого товара фиксируется место хранения (определенная полка), количество товара и его поставщик. Система поддержки заказа и учета товаров должна обеспечивать добавление информации о новом товаре, изменение или удаление информации об имеющемся товаре, хранение (добавление, изменение и удаление) информации о поставщиках, включающей в себя название фирмы, ее адрес и телефон. При помощи системы составляются заказы поставщикам. Каждый заказ может содержать несколько позиций, в каждой позиции указываются наименование товара и его количество в заказе. Система по требованию пользователя формирует и выдает на печать следующую справочную информацию:

* список всех товаров;
* список товаров, имеющихся в наличии;
* список товаров, количество которых необходимо пополнить;
* список товаров, поставляемых данным поставщиком.

Первым шагом в формировании требований является идентификация опорных точек зрения. Во всех методах формирования требований, основанных на использовании точек зрения, начальная идентификация является наиболее трудной задачей. Один из подходов к идентификации точек зрения — метод "мозговой атаки", когда определяются потенци­альные системные сервисы и организации, взаимодействующие с системой. Организуется встреча лиц, участвующих в формировании требований, которые предлагают свои точки зрения. Эти точки зрения представляются в виде диаграммы, состоящей из ряда круговых областей, отображающих возможные точки зрения (рис. 4). Во время "мозговой атаки" необходимо идентифицировать потенциальные опорные точки зрения, системные сервисы, входные данные, нефункциональные требования, управляющие события и исключительные ситуации.

Следующей стадией процесса формирования требований будет идентификация опорных точек зрения (на рис. 4 показаны в виде темных круговых областей) и сервисов (показаны в виде затененных областей). Сервисы должны соответствовать опорным точкам зрения. Но могут быть сервисы, которые не поставлены им в соответствие. Это означает, что на начальном этапе "мозговой атаки" некоторые опорные точки зрения не были идентифицированы.



*Рис. 4. Диаграмма идентификации точек зрения*

В таблице 1 показано распределение сервисов для некоторых идентифицированных на рис. 4 точек зрения. Один и тот же сервис может быть соотнесен с несколькими точками зрения.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **клиент** | **покупатель** | **постоянный покупатель** | **товар** | **поставщик** | **продавец** | **администратор** |
| Проверка наличия товара | Занесение в список постоянных клиентов | Получение скидки | Прием товара | Занесение в базу данных (название, адрес, телефон и т.д.) | Продажа товара | Доступ к базе данных |
| Покупка товара |  | Получение информацию о новых поступлениях | Занесение в базу данных (данные о поставщике, кол-ве, месте хранения и.д.) |  | Печать чека | Проверка статистики |
| Получение чека |  |  | Назначение цены |  | Доступ к каталогу | Переопределение цены |
| Заказ товара |  |  | Переопределение цены |  | Проверка наличия товара | Оформление заказа поставщику |
| Занесение покупателя и суммы покупки в базу данных |  |  | «Покупаемый» или «непокупаемый» товар |  | Оформление заказа покупателю | Печать заказа |

Таблица 1 - Сервисы, соотнесенные с точками зрения

 Информация, извлеченная из точек зрения, используется для заполнения форм шаб­лонов точек зрения и организации точек зрения в иерархию наследования. Это позволяет увидеть общие точки зрения и повторно использовать информацию в иерархии наследо­вания. Сервисы, данные и управляющая информация наследуются подмножеством точек зрения. На рис. 5 показана часть иерархии точек зрения для системы поддержки заказа и учета товаров.

#### **http://unesco.kemsu.ru/study_work/method/po/UMK/lab_pract/lab02.5.jpg**

#### **Рис. 5. Иерархия точек зрения**

**Аттестация требований**

Аттестация должна продемонстрировать, что требования действительно определяют ту систему, которую хочет иметь заказчик. Проверка требований важна, так как ошибки в спе­цификации требований могут привести к переделке системы и большим затратам, если будут обнаружены во время процесса разработки системы или после введения ее в эксплуатацию. Стоимость внесения в систему изменений, необходимых для устранения ошибок в требова­ниях, намного выше, чем исправление ошибок проектирования или кодирования. Причина в том, что изменение требований обычно влечет за собой значительные изменения в систе­ме, после внесения которых она должна пройти повторное тестирование.

Во время процесса аттестации должны быть выполнены различные типы проверок требований.

1. *Проверка правильности требований.*Пользователь может считать, что система необ­ходима для выполнения некоторых определенных функций. Однако дальнейшие размышления и анализ могут привести к необходимости введения дополнительных или новых функций. Системы предназначены для разных пользователей с различными потребностями, и поэтому набор требований будет представлять собой неко­торый компромисс между требованиями пользователей системы.
2. *Проверка на непротиворечивость.*Спецификация требований не должна содержать про­тиворечий. Это означает, что в требованиях не должно быть противоречащих друг другу ограничений или различных описаний одной и той же системной функции.
3. *Проверка на полноту.*Спецификация требований должна содержать требования, ко­торые определяют все системные функции и ограничения, налагаемые на систему.
4. *Проверка на выполнимость.*На основе знания существующих технологий требования должны быть проверены на возможность их реального выполнения. Здесь также проверяются возможности финансирования и график разработки системы.

Существует ряд методов аттестации требований, которые можно использовать совместно или каждый в отдельности.

1. *Обзор требований.*Требования системно анализируются рецензентами.
2. Прототипирование. На этом этапе прототип системы демонстрируется конечным пользователям и заказчику. Они могут экспериментировать с этим прототипом, чтобы убедиться, что он отвечает их потребностям.
3. *Генерация тестовых сценариев.*В идеале требования должны быть такими, чтобы их реализацию можно было протестировать. Если тесты для требований разрабаты­ваются как часть процесса аттестации, то часто это позволяет обнаружить пробле­мы в спецификации. Если такие тесты сложно или невозможно разработать, то обычно это означает, что требования трудно выполнить и поэтому необходимо их пересмотреть.
4. *Автоматизированный анализ непротиворечивости.*Если требования представлены в виде структурных или формальных системных моделей, можно использовать инструментальные CASE-средства для проверки непротиворечивости моделей. Для автоматизированной проверки непротиворечивости необходимо построить базу данных требований и затем проверить все требования в этой базе данных. Анализатор требований готовит отчет обо всех обна­руженных противоречиях.

**Пользовательские и системные требования**

На основании полученных моделей строятся пользовательские требования, т.е. как было сказано в начале описание на естественном языке функции, выполняемых системой, и ограничений, накладываемых на неё.

Пользовательские требования должны описывать внешнее поведение системы, основные функции и сервисы предоставляемые системой, её нефункциональные свойства. Необходимо выделить опорные точки зрения и сгруппировать требования в соответствии с ними. Пользовательские требования можно оформить как простым перечислением, так и используя нотацию вариантов использования.

Далее составляются системные требования. Они включат в себя:

1. Требования к архитектуре системы. Например, число и размещение хранилищ и серверов приложений.
2. Требования к параметрам оборудования. Например, частота  процессоров серверов и клиентов, объём хранилищ, размер оперативной и видео памяти, пропускная способность канала и т.д.
3. Требования к параметрам системы. Например, время отклика на действие пользователя, максимальный размер передаваемого файла, максимальная скорость передачи данных, максимальное число одновременно работающих пользователей и т.д.
4. Требования к программному интерфейсу.
5. Требования к структуре системы. Например, Масштабируемость, распределённость, модульность, открытость.

* масштабируемость – возможность распространения системы на большое количество машин, не приводящая к потере работоспособности и эффективности, при этом способность системы наращивать свою мощность должна определяться только мощностью соответствующего аппаратного обеспечения.
* распределенность - система должна поддерживать распределённое хранение данных.
* модульность - система должна состоять из отдельных модулей, интегрированных между собой.
* открытость - наличие открытых интерфейсов для возможной доработки и интеграции с другими системами.

1. Требования по взаимодействию и интеграции с другими системами. Например, использование общей базы данных, возможность получения данных из баз данных определённых систем и т.д.

**Техническое задание**

Техническое задание содержит следующие разделы, которые могут быть разделены на подразделы:

1) общие сведения;

2) назначение и цели создания (развития) системы;

3) характеристика объектов автоматизации;

4) требования к системе;

5) состав и содержание работ по созданию системы;

6) порядок контроля и приемки системы;

7) требования к составу и содержанию работ по подготовке объекта автоматизации к вводу системы в действие;

8) требования к документированию;

9) источники разработки.

**4.     Порядок выполнения работы**

1. Изучить предлагаемый теоретический материал.
2. Составить подробное описание информационной системы.
3. На основании описания системы провести анализ осуществимости. В ходе анализа ответить на вопросы:

* *Что произойдет с организацией, если система не будет введена в эксплуатацию?*
* *Какие текущие проблемы существуют в организации и как новая система поможет их решить?*
* *Каким образом система будет способствовать целям бизнеса?*
* *Требует ли разработка системы технологии, которая до этого не использовалась в организации?*

Результатом анализа должно явиться заключение о возможности реализации проекта.

1. Распределить роли в группе  (руководитель проекта-разработчик, системный аналитик-разработчик, тестер-разработчик).
2. Заполнить разделы плана:

* *Введение*
* *Организация выполнения проекта*
* *Анализ рисков*

Разделы должны содержать рекомендации относительно разработки системы, базовые предложения по объёму требуемого бюджета, числу разработчиков, времени и требуемому программному обеспечению.

1. Построить опорные точки зрения на основании метода VORD для формирования и анализа требований. Результатом должны явиться две диаграммы: диаграмма идентификации точек зрения и диаграмма иерархии точек зрения.
2. Составить информационную модель будущей системы, включающую в себя описание основных объектов системы и взаимодействия между ними. На основании полученной информационной модели и диаграмм идентификации точек зрения, диаграмма иерархии точек зрения сформировать требования пользователя и системные требования.
3. Провести аттестацию требований, указать какие типы проверок выбрали.
4. На основании описания системы, информационной модели, пользовательских и  системных требований составить техническое задание на создание программного обеспечения (пример см. приложение).
5. Составить отчет о проделанной работе.

**5.     Содержание отчета**

В отчете следует указать:

1. Цель работы
2. Техническое задание на создание системы.
3. Диаграмма идентификации точек зрения и диаграмма иерархии точек зрения.
4. Роли участников группы разработки ПО.
5. Заключение (выводы)

**Варианты заданий**

1. **Логическая игра «Мозаика»**

Разрабатываемое приложение представляет собой программную реализацию популярной игры «Мозаика» («Puzzle»), цель которой заключается в создании игроком полного изображения путем упорядочивания его фрагментов последовательным перемещением и соединением соответствующих сторон. Приложение должно обеспечивать:

* выбор пользователем произвольного исходного изображения и задание параметров его разбиения на фрагменты;
* сохранение в любой момент времени текущей игры в файл и загрузку сохраненной игры для продолжения, а также создание новой игры;
* просмотр исходного изображения во время игры, а также поиск подходящих друг к другу фрагментов (функция подсказки);
* возможность перемещать фрагменты за пределы рабочего поля или в другое окно и обратно для улучшения наглядности.
* подсчет и индикацию количества правильных соединений фрагментов и определение момента полного сбора изображения.

Следует учесть, что при создании новой игры все фрагменты должны быть разъединены и размещены беспорядочно в области рабочего поля или в области, отведенной специально для фрагментов. При использовании отдельной области для отображения фрагментов может быть применена ее прокрутка и изменение масштаба.

1. **Логическая игра «Пятнашки»**

Разрабатываемое приложение представляет собой программную реализацию известной игры «Пятнашки», цель которой заключается в упорядочивании игроком фишек путем их последовательного перемещения на соседнее пустое место. Приложение должно обеспечивать:

* возможность задания размера поля от 4х4 до 10х10 фишек;
* правильное перемешивание фишек;
* отмену сделанных ходов;
* подсчет и индикацию количества ходов и правильно размещенных фишек;
* фиксацию имени и результата чемпиона для каждого допустимого размера поля.

Следует учесть, что при произвольной расстановке фишек на поле в 50% случаев становится невозможным их упорядочивание, поэтому перемешивание фишек должно осуществляться по тем же правилам, что и их перемещение игроком. Кроме того, после перемешивания фишек в начале игры не более 10% от их количества могут находиться на своих местах.

1. **Приложение расчета характеристик матриц**

Приложение предназначено для выполнения некоторых операций матричной арифметики. Приложение должно обеспечивать:

* создание, редактирование и отображение матрицы;
* чтение матрицы из файла в формате .csv и ее запись в файл;
* выполнение основных операций матричной арифметики, отображение результатов и их запись в текстовый файл;
* одновременная работа с несколькими матрицами.

Формат .csv (значения, разделенные запятыми) является стандартным и позволяет записать в текстовый файл структурированные данные. В данном случае каждая строка может представлять соответствующую строку матрицы, а каждое значение в строке, отделенное запятыми, соответствовать элементу строки матрицы.

К основным операциям матричной арифметики можно отнести:

транспонирование, нахождение определителя, нахождение ранга, нахождение определителей минорных матриц (результат может быть представлен в виде отдельной матрицы), приведение к треугольной матрице, нахождение корней системы линейных уравнений, нахождение суммы элементов матрицы и среднего значения. Потребительские свойства приложения могут быть улучшены за счет использования контекстного меню, панелей инструментов и системного буфера обмена (clipboard).

1. **Логическая игра «Точки»**

Данное приложение является реализацией известной логической игры. В данной реализации предусматривается игра двух игроков на квадратном поле фиксированного размера. Цель игры – заполнить максимальное количество клеток игрового поля своими фишками. Игроки выполняют ходы по очереди. Ход игрока заключается в произвольной установке линии на границе двух ячеек, причем, если какая-либо ячейка оказывается со всех четырех сторон обрамлена линиями, то она отмечается фишкой данного игрока, а самому игроку предоставляется дополнительный ход, и т.д. Игра заканчивается, когда все игровое поле оказывается заполнено фишками. Выигрывает тот игрок, чьих фишек на поле больше на момент окончания игры.

Приложение должно обеспечивать:

* начало новой игры на чистом поле;
* сохранение в любой момент времени текущей игры в файл и загрузку сохраненной игры для продолжения, а также создание новой игры;
* проверку соответствия действий игроков правилам игры;
* игру между двумя пользователями по сети;
* условия окончания игры.

Потребительские свойства приложения могут быть улучшены за счет использования контекстного меню и панелей инструментов

1. **Игра «Морской бой»**

Разрабатываемое приложение представляет собой программную реализацию известной логической игры. Приложение должно обеспечивать:

* расстановку «кораблей» на игровом поле 10х10.
* сохранение в любой момент времени текущей игры в файл и загрузку сохраненной игры для продолжения, а также создание новой игры;
* игру между двумя соперниками по сети;
* фиксацию имен противников и число побед.

**Приложение.**

**ТЕХНИЧЕСКОЕ   ЗАДАНИЕ**

**на разработку ИС «Система»**

# Общие сведения

## **1.1. Наименование системы**

Аналитическая информационная система «Система».

# 2.1. Назначение и цели создания системы

Система «Система» предназначена для информационного обеспечения процессов, которые происходят на кафедре связанных с учебно-методической, научной, общественной, организационно-методической и воспитательной работой.

# Характеристика объектов информатизации

## **3.1. Краткое описание работы кафедры**

К основным направлениям работы кафедры относятся:

·         Учебно-методическая работа;

·         Научная работа;

·         Организационно-методическая работа;

·         Работа со студентами заочниками;

·         Общественная работа;

·         Воспитательная работа.

…

## **3.2. Описание объектов информатизации**

К основным объектам информатизации системы относятся:

*Кафедра*

·         Наименование кафедры

·         Факультет, к которому относится кафедра

·         Веб-сайт кафедры

·         Заведующий кафедрой

…

### ***3.2.1. Учебно-методическая работа***

*План учебно-методической работы кафедры*

·         Учебный год

·         Заведующий кафедрой, составивший план

·         Кафедра

*Тема для учебно-методической работы*

·         Названия работ

·         Сроки исполнения

·         Ответственные за выполнение темы

…

# Требования к информационной системе

## **4.1. Базовые принципы разработки подсистем**

При проектировании и разработке подсистем должны использоваться следующие базовые принципы:

·         Исключение дублирования ввода информации и повышение ее достоверности, за счет отождествления ранее введенной информации;

…

Система должна удовлетворять следующим требованиям:

·         Пользовательский интерфейс системы должен быть сформирован в соответствии с навыками и профилем пользователей;

…

Система должна содержать:

·         Средства поиска информации;

…

Выбор прикладного программного обеспечения системы должен удовлетворять следующим критериям:

·         Интеграция с базами данных, поддерживающих Web-технологии;

…

## 4.2. Требования к архитектуре системы.

Архитектура системы «Система» является трехзвенной. В качестве клиентского приложения выступает стандартный веб-браузер.

…

## 4.3. Требования к способам и средствам связи для информационного обмена между компонентами (модулями) Системы

Подсистемы должны взаимодействовать в пределах единой компьютерной сети (Интернет/Интранет), в которой происходит весь обмен информацией.

…

## 4.4. Требования к характеристикам взаимосвязей системы со смежными системами

Смежными системами для информационной системы «Система» являются: «Система2»,

…

**4.5. Требования к режимам функционирования подсистемы**

Разрабатываемая система должна функционировать 24 часа в сутки, 365 дней в году…

…

## 4.6. Требования к пользователям

Система подразумевает четыре типа пользователя:

·         Сотрудник – имеет доступ к просмотру общих данных по своей кафедре, а также к просмотру и редактированию личных данных, имеет возможность ;

…

## **4.7. Требования по эргономике и технической эстетике**

Основными требованиями по эргономике и технической эстетике является адекватность времени реакции модулей системы на сложность запроса пользователя к базам данных:

·      При выполнении стандартных запросов пользователь должен работать с системой в реальном режиме времени;

…

## 4.8. Требования к численности и квалификации персонала системы и режиму его работы.

*Квалификация персонала, порядок его подготовки и контроль знаний и навыков.*

…

## 4.9. Требования к защите информации от несанкционированного доступа.

Разрабатываемая система должна обладать специализированной подсистемой разграничения доступа к информационным ресурсам, функционирующей на основе системы пользователей и пользовательских групп.

…

**4.10. Требования к обмену данными**

·         Обмен данными должен происходить по сети в среде Intranet/Internet с поддержкой протокола TCP/IP;

…

## 4.11. **Требования к внешней среде системы**

*Сервер баз данных или сервер приложений должен обеспечивать:*

…

## 4.12. Требования к хранению данных

База данных «Система» должна содержать следующие данные:

·         Данные о планировании учебно-методической работы;

…

## 4.13. Требования к отдельным подсистемам

### ***4.13.1. Учебно-методическая работа***

*Функции заведующего кафедрой*

·         Создание плана учебно-методической работы на учебный семестр, заполнения, редактирования и удаления данных плана;

…

# Состав и содержание работ по созданию Системы

Разработать модель БД, позволяющую хранить и обрабатывать все необходимые…

…

# Приемо-сдаточные испытания  Системы

После завершения всех работ по разработке компонентов, настройке подсистем  и

…

# Внесение корректировок в программный продукт, связанных с ошибками в Системе

Все ошибки, которые будут выявлены в работе Системы в течении 12 месяцев

…

# Тестирование

Перед сдачей Модулей и Компонент Заказчику для выявления возможных сбоев в работе

…

# Порядок контроля и приемки Системы

Для проверки выполнения заданных функций Системы, определения и проверки соответствия требованиям ТЗ количественных и (или) качественных характеристик Системы, выявления и устранения недостатков в действиях Системы и в разработанной документации, поэтапного контроля над ходом разработки должны быть проведены следующие виды испытаний:

·         Предварительные;

…

# Процедуры тестирования и контроля качества

При проведении испытаний должны использоваться следующие типы процедур тестирования и контроля качества:

·         функциональное тестирование - тестирование ПО на соответствие функциональным спецификациям;

…

# Общие требования к приемке работ

Сроки и место приемки, порядок приемки работ определяются в соответствии с настоящим ТЗ.

…

# Требования к документированию

## **12.1. Требования к проектной документации**

Состав и комплектность проектной документации должна соответствовать требованиям ГОСТ 34.201-89.

Перечень документации по созданию системы включает:

·         Описание информационного обеспечения системы