

Разработка технического задания на создание программного обеспечения.

Цель работы:

Проанализировать требования к программному обеспечению, составить и оформить техническое задание на разработку.

Основная задача лабораторной работы познакомиться с процессом разработки требований к программному обеспечению и составления технического задания на разработку программы, получение навыков использования основных методов формирования и анализа требований.

Теоретические сведения.

В процессе создания программного обеспечения специалистам приходится решать ряд сложных задач. В ряде случаев 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 функциональных возможностей и ограничений – *разработкой требований*.

Требования делят на пользовательские и системные.

Пользовательские требования – это описание функций выполняемых системой и ограничений, накладываемых на неё на естественном языке.

Системные требования – это описание архитектуры системы, требований к параметрам оборудования, программного окружения и т.д., необходимых для эффективной реализации поставленной пользователем задачи.

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

1. анализ технической возможности создания системы,

2. формирование и анализ требований,
3. специфицирование требований и создание соответствующей документации,
4. аттестация требований.

В ходе анализа технической возможности создания системы выполняются общее описание системы и ее назначения. В результате анализа формируется отчет, в котором дается четкая рекомендация о возможности продолжения процесса разработки требований проектируемой системы.

Анализ должен ответить на следующие вопросы.

1. Решает ли система задачи организации-заказчика и организации-разработчика?
2. Возможно ли реализовать систему, на основе существующих на данный момент технологии, не выходя за пределы заданной стоимости?
3. Возможно ли объединить разрабатываемую систему с другими системами, которые уже эксплуатируются?

Для ответов на поставленные вопросы нужно определить 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 развитию заказчика?
4. Требуется ли разработка системы технологии, которая до этого не использовалась в организации?

Источниками для получения необходимой информации могут быть менеджеры отделов, где система будет использоваться, разработчики программного обеспечения, знакомые с типом будущей системы, конечные пользователи и т.д.

После обработки собранной информации готовится отчет по анализу возможности создания системы. В нем даются рекомендации о

продолжении разработки системы, предлагаются изменения бюджета и графика работ по созданию системы, пересматриваются требования к системе.

На втором этапе выполняется формирование и анализ требований (рис.1). Каждая организация использует собственный вариант этой модели, зависящий от опыта работы коллектива разработчиков, типа разрабатываемой системы, используемых стандартов и т.д.

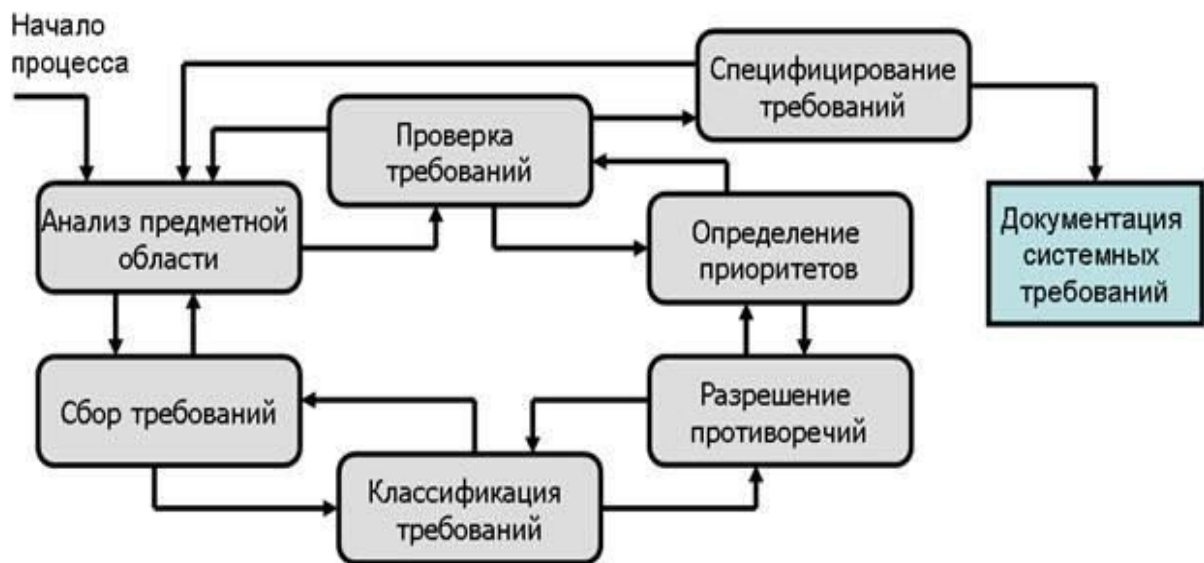


Рис.1. Процесс формирования и анализа требований

В процессе формирования и анализа требований выполняют следующие работы:

1. *Анализ предметной области.* Аналитики изучают предметную область, в которой будет эксплуатироваться система.
2. *Сбор требований.* Т.е. взаимодействия с лицами, формирующими требования и продолжение анализа предметной области.
3. *Классификация требований.* Преобразование набора требований в логически связанные группы.
4. *Разрешение противоречий.* Так как требования многочисленных лиц, занятых в процессе формирования требований, могут быть противоречивыми, необходимо определить и разрешить эти противоречия.

5. *Назначение приоритетов.* В любом наборе требований некоторые из них будут более важны. Поэтому совместно с лицами, формирующими требования, определяются наиболее важные требования.

6. *Проверка требований.* Определяется полнота, последовательность и непротиворечивость выделенных требований.

Процесс формирования и анализа требований циклический, с обратной связью. Цикл начинается с анализа предметной области и заканчивается проверкой требований. Понимание требований предметной области улучшается в каждом цикле процесса формирования требований.

Существуют два основных подхода к формированию требований:

- метод, основанный на множестве опорных точек зрения;
- сценарии.

Метод с использованием различных *опорных* точек зрения к разработке требований признает различные точки зрения на проблему и использует для построения и организации как процесса формирования требований, так и непосредственно самих требований.

Точки зрения можно трактовать следующим образом.

- *Источник информации о системных данных.* На основе опорных точек зрения строится модель создания и использования данных в системе. В процессе формирования требований отбираются все такие точки зрения, которые будут использованы при работе системы, а также способы обработки этих точек зрения (данных).

- *Структура представлений.* Точки зрения рассматриваются как особая часть модели системы. На основе различных точек зрения могут разрабатываться модели "сущность-связь", модели конечного автомата и т.д.

- *Получатели системных сервисов* Точки зрения являются внешними относительно системы получателями сервисов, они помогают определить данные, необходимые для выполнения системных сервисов.

Наиболее эффективным подходом к анализу систем является использование внешних опорных точек зрения, на основе которого

разработан метод VORD (Viewpoint-Oriented Requirements Definition — определение требований на основе точек зрения). Основные этапы метода VORD представлены на рисунке 2.:

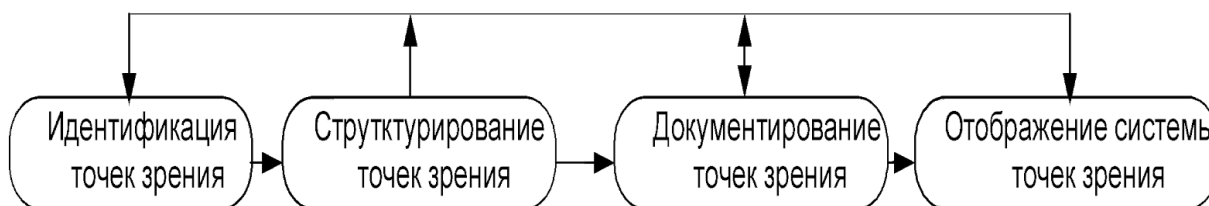


Рис.2. Метод VORD

Идентификация точек зрения, получающих системные сервисы, и идентификация сервисов, соответствующих каждой точке зрения.

Структурирование точек зрения — создание иерархии сгруппированных точек зрения. Общесистемные сервисы предоставляются более высоким уровням иерархии и наследуются точками зрения низшего уровня.

Документирование опорных точек зрения, которое заключается в точном описании идентифицированных точек зрения и сервисов.

Отображение системы точек зрения, которая показывает системные объекты, определенные на основе информации, заключенной в опорных точках зрения.

Рассмотрим использование метода VORD для анализа требований к системе поддержки заказа и учета товаров в магазине. Пусть для каждого товара фиксируется место хранения, количество товара и его поставщик. Система учета заказов и товаров должна обеспечивать добавление информации о новом товаре, изменение или удаление информации об имеющемся товаре, хранение (добавление, изменение и удаление) информации о поставщиках (название фирмы, ее адрес и телефон). Система позволяет составлять заказы поставщикам. Каждый заказ может содержать несколько позиций, в каждой из которых указывается наименование товара и его количество в заказе. Система по требованию пользователя формирует и выдает на печать следующую информацию:

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

При формировании требований необходимо идентифицировать опорные точки зрения. Один из подходов к идентификации точек зрения — метод "мозгового штурма", при котором определяют потенциальные системные сервисы и организации, взаимодействующие с системой. В ходе "мозгового штурма" идентифицируют потенциальные опорные точки зрения, системные сервисы, входные данные, нефункциональные требования, управляющие события и исключительные ситуации.

На рисунке 3 приведена диаграмма идентификации точек зрения и сервисов. Сервисы должны соответствовать опорным точкам зрения. Но могут быть сервисы, которые не поставлены им в соответствие. Это означает, что на начальном этапе "мозгового штурма" некоторые опорные точки зрения не были идентифицированы.

Распределение сервисов для некоторых идентифицированных на рисунке 3 точек зрения приведено в таблице 1. Один и тот же сервис может быть соотнесен с несколькими точками зрения.

Информация, извлеченная из точек зрения, используется для заполнения форм шаблонов точек зрения. После чего точки зрения организуются в иерархию наследования, что позволяет увидеть общие точки зрения и повторно использовать информацию. Сервисы, данные и управляющая информация наследуются подмножеством точек зрения.

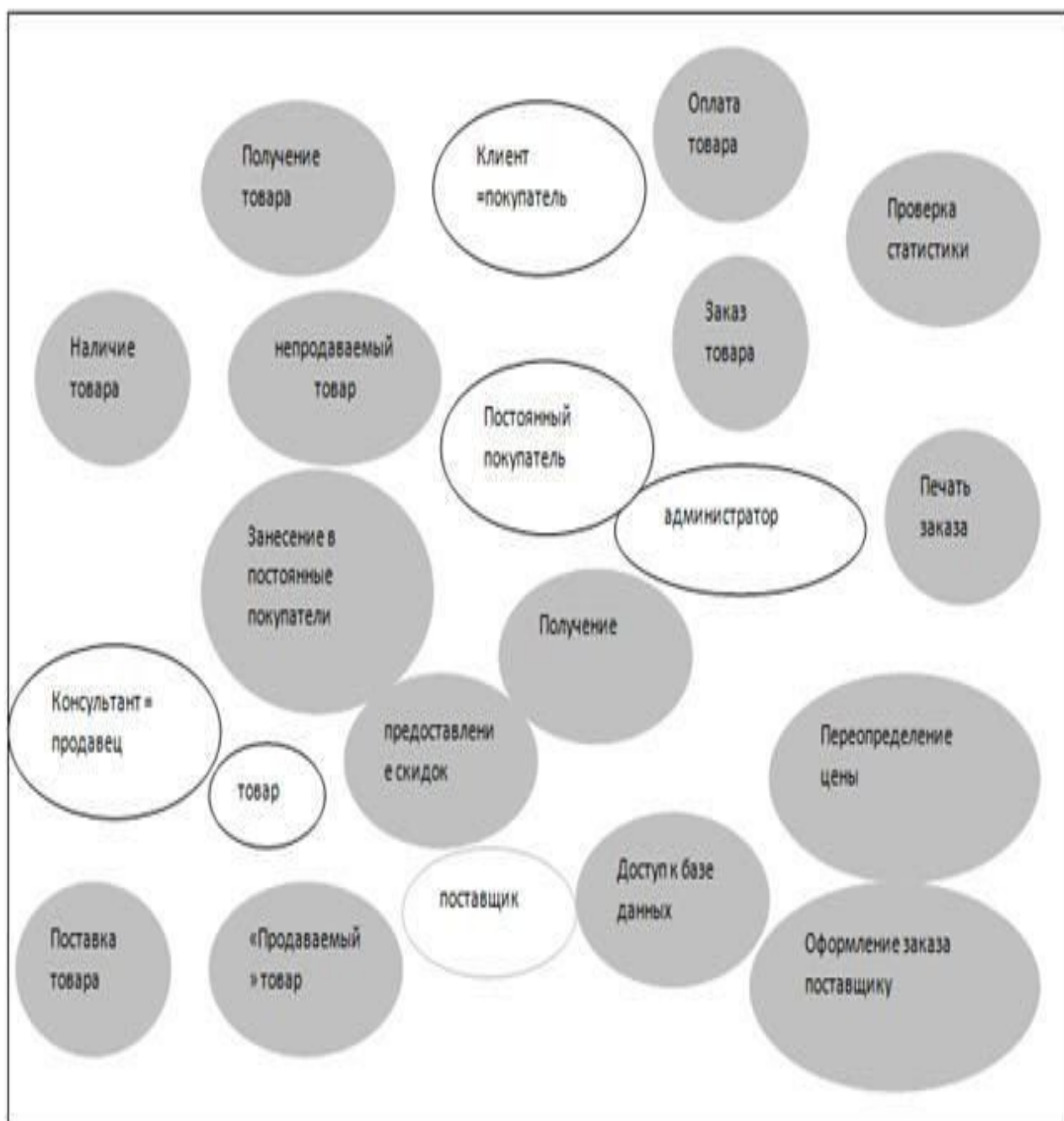


Рис.3. Диаграмма идентификации точек зрения

На рисунке 4 показана часть иерархии точек зрения для системы учета заказов и товаров.

Таблица 1

Сервисы, соотнесенные с точками зрения

Клиент	Постоянный покупатель	Товар	Поставщик	Продавец	Администратор
Проверка наличия товара	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



Рис.4. Иерархия точек зрения

Сценарии особенно полезны для детализации уже сформулированных требований, так как описывают варианты взаимодействия пользователя с системой. Каждый сценарий описывает одно или несколько возможных интерактивных действий.

Сценарий начинается с общего описания, затем постепенно детализируется для создания полного описания взаимодействия пользователя с системой.

В большинстве случаев сценарий включает следующее:

- Описание состояния системы в начале сценария.
- Описание состояния системы после завершения сценария.
- Описание нормального протекания событий.
- Описание исключительных ситуаций и способов их обработки
- Информацию относительно других действий, которые можно осуществлять во время выполнения сценария.

Таким образом, сценарии содержат описание потоков данных, системных операций и исключительных ситуаций, которые могут возникнуть

Для понимания и формирования социальных и организационных аспектов эксплуатации системы используется этнографический подход к формированию системных требований, представленный на рисунке 5.

Разработчик требований погружается в рабочую среду, в которой будет использоваться система, наблюдает и протоколирует реальные действия,

выполняемые пользователями системы. Это помогает обнаружить неявные требования к системе, которые отражают реальные аспекты ее эксплуатации, а не формальные умозрительные процессы.

Этнографический подход позволяет детализировать требования для критических систем, чего не всегда можно добиться другими методами разработки требований. Однако, поскольку этот метод не может охватить все требования предметной области и требования организационного характера так как он ориентирован на конечного пользователя.



Рис.5. Процесс разработки требований согласно этнографическому подходу

То что требования действительно определяют систему, которую хочет иметь заказчик демонстрирует аттестация требований, Проверка требований очень важна, так как ошибки в спецификации требований могут привести к переделке всей системы и большим затратам, если будут обнаружены в процессе разработки или после введения 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.

Во время процесса аттестации выполняются 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.

3. *Проверка полноты.* Спецификация требований должна содержать требования, которые определяют все системные функции и ограничения, налагаемые на систему.

4. *Проверка выполнимости.* На основе знания существующих технологий требования проверяются на возможность их реального выполнения, а так же возможность финансирования и соблюдения графика разработки системы.

Существует следующие методы аттестации требований, которые можно использовать как совместно, так и каждый в отдельности.

1. *Обзор требований.* Требования системно анализируются рецензентами.

2. *Прототипирование.* Прототип системы демонстрируется конечному пользователю или заказчику с целью подтверждения того, что он отвечает потребностям.

3. *Генерация тестовых сценариев.* Требования должны быть такими, чтобы их реализацию можно было протестировать. Если тесты для требований разрабатываются как часть процесса аттестации, то возможно обнаружить проблемы в спецификации. Если такие тесты сложно или невозможно разработать, это означает, что требования трудно выполнить и 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 строят базу данных требований и затем проверяют все требования в этой базе данных. Анализатор требований готовит отчет обо всех обнаруженных противоречиях.

На основании полученных моделей строятся пользовательские требования, описывающие функции, выполняемые системой, и ограничений, накладываемых на неё на естественном языке.

Пользовательские требования должны описывать внешнее поведение системы, основные функции и сервисы предоставляемые системой, её нефункциональные свойства. Пользовательские требования можно оформить простым перечислением.

Затем составляются системные требования:

1. *Требования к архитектуре системы.* Например, число и размещение хранилищ и серверов приложений.

2. *Требования к техническому обеспечению.* Например, частота процессоров серверов и клиентов, требуемый объём памяти для хранения данных, 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 (возможность распространения системы на большое количество машин, не приводящая к потере работоспособности и эффективности), распределённость (поддержка распределённого хранения данных),

модульность, открытость (наличие открытых интерфейсов для возможной доработки и интеграции с другими системами).

6. *Требования по взаимодействию и интеграции с другими системами.* Например, использование общих баз данных, получение данных из баз данных других систем и т.д.

Техническое задание оформляют в соответствии с ГОСТ 34.602—89.

Техническое задание должно содержать следующие разделы:

- введение;
- наименование и область применения;
- основание для разработки;
- назначение разработки;
- технические требования к программе или программному изделию;
- технико-экономические показатели;
- стадии и этапы разработки;
- порядок контроля и приемки;
- приложения.

В зависимости от особенностей программы или программного изделия допускается уточнять содержание разделов, вводить новые разделы или объединять отдельные из них. При необходимости допускается в техническое задание включать приложения.

В соответствии с ГОСТ содержание разделов определяется следующим образом.

Введение должно включать краткую характеристику области применения программы или программного продукта, а также объекта (например, системы), в котором предполагается их использовать. Основное назначение введения — продемонстрировать актуальность данной разработки и показать, какое место эта разработка занимает в ряду подобных.

В разделе «Наименование и область применения» указывают наименование, краткую характеристику области применения программы или

программного изделия и объекта, в котором используют программу или программное изделие.

В разделе «Основание для разработки» должны быть указаны:

- документ (документы), на основании которых ведется разработка. Таким документом может служить план, приказ, договор и т. п.;
- организация, утвердившая этот документ, и дата его утверждения;
- наименование и (или) условное обозначение темы разработки.

В разделе «Назначение разработки» должно быть указано функциональное и эксплуатационное назначение программы или программного изделия.

Раздел «Технические требования к программе или программному изделию» должен содержать следующие подразделы:

- требования к функциональным характеристикам - должны быть указаны требования к составу выполняемых функций, организации входных и выходных данных, временным характеристикам и т. п.;
- требования к надежности - должны быть указаны требования к обеспечению надежного функционирования (обеспечение устойчивого функционирования, контроль входной и выходной информации, время восстановления после отказа и т. п.);
- условия эксплуатации - должны быть указаны условия эксплуатации (температура окружающего воздуха, относительная влажность и т. п. для выбранных типов носителей данных), при которых должны обеспечиваться заданные характеристики, а также вид обслуживания, необходимое количество и квалификация персонала;
- требования к составу и параметрам технических средств - указывают необходимый состав технических средств с указанием их технических характеристик;
- требования к информационной и программной совместимости - должны быть указаны требования к информационным структурам на входе и

выходе и методам решения, исходным кодам, языкам программирования. При необходимости должна обеспечиваться защита информации и программ;

- требования к маркировке и упаковке в общем случае указывают требования к маркировке программного изделия, варианты и способы упаковки;
- требования к транспортированию и хранению должны быть указаны для программного изделия условия транспортирования, места хранения, условия хранения, условия складирования, сроки хранения в различных условиях;
- специальные требования.

В разделе «Технико-экономические показатели» должны быть указаны: ориентировочная экономическая эффективность, предполагаемая годовая потребность, экономические преимущества разработки по сравнению с лучшими отечественными и зарубежными образцами или аналогами.

В разделе «Стадии и этапы разработки» устанавливают необходимые стадии разработки, этапы и содержание работ (перечень программных документов, которые должны быть разработаны, согласованы и утверждены), а также, как правило, сроки разработки и определяют исполнителей.

В разделе «Порядок контроля и приемки» должны быть указаны виды испытаний и общие требования к приемке работы.

В приложениях к техническому заданию при необходимости приводят:

- перечень научно-исследовательских и других работ, обосновывающих разработку;
- схемы алгоритмов, таблицы, описания, обоснования, расчеты и другие документы, которые могут быть использованы при разработке;
- другие источники разработки.

В случаях, если какие-либо требования, предусмотренные техническим заданием, заказчик не предъявляет, следует в соответствующем месте указать «Требования не предъявляются».

Порядок выполнения работы

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;
2. Составить информационную модель будущего программного обеспечения, включающую в себя описание основных объектов системы и взаимодействия между ними;
3. Определить пользовательские требования, четко описывающие будущий функционал системы;
4. Определить системные требования, включающие требования к структуре, программному интерфейсу, технологиям разработки, общие требования к систем
5. Разработать техническое задание на программный продукт