**09.03.2020**

**Проведение предпроектных исследований. Разработка технического задания.**

Осуществляется с целью параметризации проекта создания ИС. Сначала выявляются все материальные, финансовые и временные ресурсы для выполнения необходимых проектных работ. Это этап сбора необходимых данных и их анализа.

К задачам предпроектного обследования относятся следующие:

1. Исследование основных бизнес-процессов на объекте автоматизации, где предполагается внедрение системы организационно-управленческого типа, либо процессов обработки инженерной и (или) научной информации, если основное назначение системы заключается в автоматизации именно этих процедур. При этом выявляются:

• процессы, содержащие избыточное количество шагов между начальной и конечной точками;

• процессы, включающие дублирующие процедуры;

• процессы, выполняемые по морально устаревшим схемам, т. е. «по инерции», хотя производственной или управленческой необходимости в них уже нет;

• процессы, которые необходимо структурно регламентировать в графическом виде и (или) описать в форме документов;

• процессы, требующие унификации.

2. Выявление тех особенностей в деятельности объекта автоматизации, которые являются уникальными и поэтому обязательно должны поддерживаться в условиях АСОИУ.

3. Формирование совокупности задач, которые будут реализованы в АСОИУ (функциональность АСОИУ).

4. Определение бюджета и длительности проекта (зависит от объема функциональности).

5. Оценка работ по интеграции проектируемой АСОИУ с уже находящимися в эксплуатации на объекте автоматизации автоматизированными системами разных типов.

6. Оценка работ по обеспечению взаимодействия проектируемой системы с внешними по отношению к объекту автоматизации системами.

7. Выявление потребностей в количественном и качеством составе аналитической информации для всех уровней управления, а также информации, необходимой для принятия решений по результатам обработки разнообразной информации, которая относится к задачам соответствующей научно-исследовательской деятельности.

8. Выявление замечаний со стороны пользователей разного профессионально-квалификационного и управленческого уровней к качеству и количеству используемой в настоящее время информации для решения текущих задач.

9. Оценка возможных рисков, например:

• часто меняющиеся требования заказчика, что может обусловливаться неумением их корректно сформулировать и недостаточной компетентностью экспертов (исключение представляют гибкие технологии разработки ПО);

• недостаточный бюджет;

• жесткие временные рамки проекта;

• слишком большой объем реализуемой функциональности при заданных сроках проекта и его бюджете;

• использование недостаточно апробированных информационных технологий;

• недостаточные квалификация и опыт разработчиков.

Предпроектное обследование объекта автоматизации необходимо проводить в следующих случаях:

• создание АСОИУ по «индивидуальному заказу»;

• внедрение готовой системы (исследовать, насколько бизнес- процессы на объекте автоматизации соответствуют тем, поддержание которых обеспечивается приобретаемым программным решением);

• внедрение дополнительных модулей «родной» системы или «иногородней»;

• функциональная и (или) технологическая модернизация существующей системы;

• внедрение дополнительных автоматизированных систем к тем, которые уже функционируют на объекте автоматизации.

По результатам предпроектного обследования должны быть:

• предварительно оценены необходимые для реализации проекта ресурсы и сроки его выполнения;

• графически представлены и критически проанализированы существующие процессы или бизнес-процессы (с использованием, например, UML-диаграмм);

• сформулированы функционально-технологические требования к АСОИУ, т. е. определена совокупность задач и их информационные взаимосвязи как внутри проектируемой системы, так и при ее взаимодействии с внешними системами.

С точки зрения оптимизации информационной поддержки бизнес- процессов в условиях будущей АСОИУ могут быть выработаны рекомендации по их изменению, включая элементы реинжиниринга, например:

• объединение нескольких «мелких» или смежных работ процесса в одну;

• выполнение процессов в порядке, соответствующем бизнес- логике той деятельности, за которую отвечает подразделение (иногда процессы в силу разных причин бывают неоправданно «раздутыми» по количеству шагов);

• предложение различных версий выполнения процесса, способствующих сокращению его длительности и трудоемкости, а также более четкой регламентации сути самого процесса;

• изменение места выполнения процесса, руководствуясь его экономической и логической целесообразностью.

Цели и задачи предпроектного обследования являются основой для выполнения самого сложного и важного процесса «Формулирование требований к будущей системе» или Техническое задание.

Под разработкой требований авторы понимают такую последовательность действий:

* выявление;
* анализ;
* документирование;
* утверждение.

Процесс обследования объекта автоматизации в контексте формулирования требований к проектируемой системе заканчивается тем, что аналитик должен достаточно хорошо представлять информационно технологическое обеспечение деятельности этого объекта.

***Техническое задание***

Техническое задание - это документ, определяющий цели, требования и основные исходные данные, необходимые для разработки автоматизированной системы управления.

При разработке технического задания необходимо решить следующие задачи:

* установить общую цель создания ИС, определить состав подсистем и функциональных задач;
* разработать и обосновать требования, предъявляемые к подсистемам;
* разработать и обосновать требования, предъявляемые к информационной базе, математическому и программному обеспечению, комплексу технических средств (включая средства связи и передачи данных);
* установить общие требования к проектируемой системе;
* определить перечень задач создания системы и исполнителей;
* определить этапы создания системы и сроки их выполнения;
* провести предварительный расчет затрат на создание системы и определить уровень экономической эффективности ее внедрения.

В зависимости от условий создания системы возможны различные совмещения функций заказчика, разработчика, поставщика и других организаций, участвующих в работах по созданию ИС.

В общем случае содержание технического задания включает следующие разделы:

1. Введение;
2. Основание для разработки;
3. Назначение разработки;
4. Технические требования к программе или программному изделию;
5. Технико-экономические показатели;
6. Стадии и этапы разработки;
7. Порядок контроля и приёмки;
8. Приложения.

В зависимости от вида, назначения, специфических особенностей объекта автоматизации и условий функционирования системы допускается оформлять разделы ТЗ в виде приложений, вводить дополнительные, исключать или объединять подразделы ТЗ.

**10.03.2020**

**Проведение тестирования алгоритма и программного продукта**

***Тестирование и его результаты***

Тестирование программного продукта проводят как на этапах его создания, так и на этапах технического сопровождения и дальнейшего развития. Виды тестирования программных продуктов и их краткое описание я постараюсь представить в этом материале.

Тестирование – самая популярная методика повышения качества и выявление и устранения дефектов, подкрепленная многими исследованиями и богатым опытом разработки коммерческих приложений. Существует множество видов тестирования: одни обычно выполняют сами разработчики, а другие – специализированные группы. Виды тестирования перечислены ниже:

Блочным тестированием называют тестирование полного класса, метода или небольшого приложения, написанного одним программистом или группой, выполняемое отдельно от прочих частей системы.

Тестирование компонента – это тестирование класса, пакета, небольшого приложения или другого элемента системы, разработанного несколькими программистами или группами, выполняемое в изоляции от остальных частей системы.

Интеграционное тестирование – это совместное выполнение двух или более классов, пакетов, компонентов или подсистем, созданных несколькими программистами или группами.

Регрессивным тестированием называют повторное выполнение тестов, направленное на обнаружение дефектов в программе, уже прошедшей этот набор тестов.

Тестирование системы – это выполнение ПО в его окончательной конфигурации, интегрированного с другими программными и аппаратными системами.

Результаты тестирования - это список выявленных несоответствий и дефектов, но, как правило, без указания их причин, хотя, опытный специалист по качеству продукции в состоянии провести первичную диагностику некорректного поведения объекта и сообщить диагноз его разработчикам.

***Тестирование на этапе создания программного продукта***

Виды тестирования на этапе [создания программного продукта](http://codingcraft.ru/programming_for_beginners.php) классифицируют по преследуемым этим процессом целям. Функциональное тестирование - проверка корректного выполнения программой заложенных в нее функций. Нефункциональное тестирование - все прочие виды испытаний, такие как тестирование производительности в различных режимах эксплуатации, тестирование эргономики пользовательского интерфейса, тестирование отказоустойчивости и т.д.

***Тестирование производительности***

Цель данного испытания — проверка соответствия параметрам, заданным на этапе проектирования системы. В ходе испытаний моделируются различные уровни нагрузки и формируются объемы данных для определения границ производительности системы.

#### **Тестирование на этапе подготовки к эксплуатации**

#### На данном этапе требуется комплексная проверка работоспособности системы. Тестируемое программное решение разворачивается на предоставленных заказчиком программно-аппаратных ресурсах и проходит предварительно согласованную программу испытаний.

#### **Модульное тестирование на этапах сопровождения**

#### Испытания, направленные на проверку дефектов и ошибок после обновления одного из системных модулей. Тестирование проводится в процессе эксплуатации информационной системы, практически не влияет на работу пользователей и требует существенно меньшего времени по сравнению с комплексным испытанием.

#### Управление качеством информационных систем — трудоемкий процесс, требующий отлаженного командного взаимодействия. Разработчики концентрируют усилия на обеспечении стабильной работы создаваемой системы, и зачастую это не позволяет им объективно оценить качество кода. Но воспользовавшись услугами независимой команды опытных тестировщиков, компания клиент сможет в кратчайшие сроки получить представление об основных системных ошибках и принять необходимые меры по совершенствованию корпоративного ПО.

**11.03.2020**

**Отладка программного обеспечения**

Отладка программы — один их самых сложных этапов разработки программного обеспечения, требующий глубокого знания:

* специфики управления используемыми техническими средствами,
* операционной системы,
* среды и языка программирования,
* реализуемых процессов,
* природы и специфики различных ошибок,
* методик отладки и соответствующих программных средств.

Отладка - это процесс локализации и исправления ошибок, обнаруженных при тестировании программного обеспечения. Локализацией называют процесс определения оператора программы, выполнение которого вызвало нарушение нормального вычислительного процесса. Исправления ошибки необходимо определить ее причину, т. е. определить оператор или фрагмент, содержащие ошибку. Причины ошибок могут быть как очевидны, так и очень глубоко скрыты.

В соответствии с этапом обработки, на котором проявляются ошибки, различают: синтаксические ошибки - ошибки, фиксируемые компилятором при выполнении синтаксического и частично семантического анализа программы; ошибки компоновки - ошибки, обнаруженные компоновщиком (редактором связей) при объединении модулей программы; ошибки выполнения - ошибки, обнаруженные операционной системой, аппаратными средствами или пользователем при выполнении программы.

***Синтаксические ошибки.***

Синтаксические ошибки относят к группе самых простых, так как синтаксис языка, как правило, строго формализован, и ошибки сопровождаются развернутым комментарием с указанием ее местоположения. Определение причин таких ошибок, как правило, труда не составляет, и даже при нечетком знании правил языка за несколько прогонов удается удалить все ошибки данного типа.

***Ошибки выполнения.***

К самой непредсказуемой группе относятся ошибки выполнения. Прежде всего они могут иметь разную природу, и соответственно по-разному проявляться. Часть ошибок обнаруживается и документируется операционной системой. Выделяют четыре способа проявления таких ошибок: • появление сообщения об ошибке, зафиксированной схемами контроля выполнения машинных команд, например, переполнении разрядной сетки, ситуации «деление на ноль», нарушении адресации и т. п.; •появление сообщения об ошибке, обнаруженной операционной системой, например, нарушении защиты памяти, попытке записи на устройства, защищенные от записи, отсутствии файла с заданным именем и т. п.; •«зависание» компьютера, как простое, когда удается завершить программу без перезагрузки операционной системы, так и «тяжелое», когда для продолжения работы необходима перезагрузка; •несовпадение полученных результатов с ожидаемыми.

***Методы отладки программного обеспечения.***

Отладка программы в любом случае предполагает обдумывание и логическое осмысление всей имеющейся информации об ошибке. Большинство ошибок можно обнаружить по косвенным признакам посредством тщательного анализа текстов программ и результатов тестирования без получения дополнительной информации. При этом используют различные методы:

1) ручного тестирования;

2) индукции;

3) дедукции;

4) обратного прослеживания.

***Метод ручного тестирования.***

Это - самый простой и естественный способ данной группы. При обнаружении ошибки необходимо выполнить тестируемую программу вручную, используя тестовый набор, при работе с которым была обнаружена ошибка. Метод очень эффективен, но не применим для больших программ, программ со сложными вычислениями и в тех случаях, когда ошибка связана с неверным представлением программиста о выполнении некоторых операций. Данный метод часто используют как составную часть других методов отладки.

***Индукции***

Метод основан на тщательном анализе симптомов ошибки, которые могут проявляться как неверные результаты вычислений или как сообщение об ошибке.

***Дедукции***

По методу дедукции вначале формируют множество причин, которые могли бы вызвать данное проявление ошибки. Затем анализируя причины, исключают те, которые противоречат имеющимся данным. Если все причины исключены, то следует выполнить дополнительное тестирование исследуемого фрагмента. В противном случае наиболее вероятную гипотезу пытаются доказать. Если гипотеза объясняет полученные признаки ошибки, то ошибка найдена, иначе - проверяют следующую причину.

***Метод обратного прослеживания.***

Для небольших программ эффективно применение метода обратного прослеживания. Начинают с точки вывода неправильного результата. Для этой точки строится гипотеза о значениях основных переменных, которые могли бы привести к получению имеющегося результата. Далее, исходя из этой гипотезы, делают предложения о значениях переменных в предыдущей точке. Процесс продолжают, пока не обнаружат причину ошибки.

**12.03.2020**

**Составление описания на программный продукт**

Основываясь на требованиях к информационной системе, я решила создать десять таблиц в базе данных:

* Заказ - данная база важна в работе мебельной фабрики. С её помощью можно следить за тем, какие заказы есть у фабрики.

ID

Дата

Этап выполнения

Заказчик

Менеджер

Стоимость

* Заказные изделия – в этой таблице описываются изделия, которые будут отправлять по заказу.

ID

ID изделия

Количество

* Изделия – в этой таблице мы указываем, какие изделия есть на фабрике, их стоимость, название и их характеристики.

Артикул

ID

Наименование

Ширина

Длина

Стоимость

* Пользователь – здесь хранятся данные заполненные пользователем ИС.

ID

Логин

Пароль

Роль

Наименование

* Склад ткани – в этой базе данных хранятся данные о тканях, которые присутствуют на складе фабрики.

ID

ID ткани

Рулон

Ширина

Длина

* Склад фурнитуры - в этой базе данных хранятся данные о фурнитуре, которые присутствуют на складе фабрики.

ID

Партия

ID фурнитуры

Количество

* Ткани - в этой таблице мы указываем, какие ткани есть на фабрике, их стоимость, название и их характеристики.

ID

Артикул

Название

Цвет

Рисунок

Состав

Ширина

Длина

Цена

* Ткани изделия - в этой таблице мы указываем, какие ткани используются для изделий

ID ткани

ID изделия

* Фурнитура - в этой таблице мы указываем, какая фурнитура есть на фабрике, их стоимость, название и их характеристики.

ID

Артикул

Наименование

Ширина

Длина

Тип

Цена

Вес

* Фурнитура изделия- в этой таблице мы указываем, какая фурнитура используется для изделий.

ID

ID фурнитуры

ID изделия

Размещение

Ширина

Длина

Поворот

Количество

Также, согласно заданию, я сделала меню авторизации для пользователя, которое состоит из:

* Кнопок входа и выхода
* Двух текстовых поля для логина и пароля.

**13.03.2020**

**Создание модулей. Выбор метода разработки модуля**

Я привязала с модульное приложение SQL к Visual Studio с помощью добавления нового элемента и добавления в библиотеку нового элемента как using System.Data.SqlClient;

**16.03.2020**

**Программирование модуля. Логическая проверка модуля**

Я программировала в Visual studio на языке C#

Первым делом, что я сделала – создала форму, на котором разместила:

* 2 button – войти и выйти
* 2 label – Логин и пароль
* 2 textbox – Для введения Логина и пароля

Потом я написала код для кнопок, чтобы при нажатии они выполняли определенные действия, например вход в систему или выход

**17.03.2020**

**Компиляция модуля. Отладка и тестирование модулей**

Компиляция — сборка программы, включающая [трансляцию](https://ru.wikipedia.org/wiki/%D0%A2%D1%80%D0%B0%D0%BD%D1%81%D0%BB%D1%8F%D1%82%D0%BE%D1%80) всех модулей программы, написанных на одном или нескольких исходных [языках программирования высокого уровня](https://ru.wikipedia.org/wiki/%D0%92%D1%8B%D1%81%D0%BE%D0%BA%D0%BE%D1%83%D1%80%D0%BE%D0%B2%D0%BD%D0%B5%D0%B2%D1%8B%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F) и/или [языке ассемблера](https://ru.wikipedia.org/wiki/%D0%AF%D0%B7%D1%8B%D0%BA_%D0%B0%D1%81%D1%81%D0%B5%D0%BC%D0%B1%D0%BB%D0%B5%D1%80%D0%B0), в эквивалентные программные модули на [низкоуровневом языке](https://ru.wikipedia.org/wiki/%D0%9D%D0%B8%D0%B7%D0%BA%D0%BE%D1%83%D1%80%D0%BE%D0%B2%D0%BD%D0%B5%D0%B2%D1%8B%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F), близком [машинному коду](https://ru.wikipedia.org/wiki/%D0%9C%D0%B0%D1%88%D0%B8%D0%BD%D0%BD%D1%8B%D0%B9_%D0%BA%D0%BE%D0%B4).

**18.03.2020**

**Отладка модуля с целью выявления логических ошибок**

При разработке программ наиболее трудоемким является этап отладки и тестирования программ. Цель тестирования, т.е. испытания программы, заключается в выявлении имеющихся в программе ошибок. Отладка представляет собой процесс обнаружения, локализации и устранения ошибок в проекте. Она занимает значительную часть рабочего времени программиста, нередко большую, чем разработка проекта.

Логические ошибки условно можно разделить на ошибки алгоритма и семантические ошибки. Причинами таких ошибок могут быть несоответствие алгоритма поставленной задаче, неправильное понимание программистом смысла (семантики) операторов языка программирования, нарушение допустимых пределов и правил представления данных, невнимательность при технической подготовке проекта к обработке на компьютере.

Для выявления ошибок служат тесты. Тест – это такой набор исходных данных, который дает результат, не вызывающий сомнений. Промежуточные и конечные результаты теста используются для контроля правильности выполнения приложения

Составление тестов – непростая задача. Тесты должны быть с одной стороны, достаточно простыми, чтобы результат легко проверялся, с другой стороны – достаточно сложными, чтобы комплексно проверить алгоритм.

Тесты составляются по схеме алгоритма до программирования, так как составление тестов помогает выявить многие ошибки в алгоритмизации.

Количество тестов и их сложность зависят от алгоритма. Комплекс тестов должен быть таким, чтобы все ветви схемы алгоритма были пройдены, по крайней мере, по одному разу. Несовпадение результатов, выдаваемых приложением с результатами тестов – признак наличия ошибок. Эти ошибки проявляются в том, что результат расчета оказывается неверным либо происходит переполнение, деление на 0 и др.

Для локализации места ошибки рекомендуется поступать следующим образом. В окне Редактора Кода установите курсор в строке перед подозрительным участком и нажмите клавишу F4 (выполнить до курсора). Выполнение приложения будет остановлено на той строке модуля, в которой был установлен курсор. Текущее значение любой переменной можно увидеть, если накрыть курсором идентификатор переменной на 1-2 сек. Нажимая клавишу F8 (пошаговое выполнение), можно построчно выполнять программу, контролируя содержимое переменных и правильность вычислений.

**19.03.2020**

**Верификация и аттестация модуля. Разработка системы тестов.**

Верификацией и аттестацией называются процессы проверки и анализа, в ходе которых проверяется соответствие программного обеспечения своей спецификации и требованиям заказчиков. Верификация и аттестация охватывают весь цикл жизни ПО они начинаются на этапе анализа требований и завершаются проверкой программного кода на этапе тестирования программной системы.

Верификация и аттестация абсолютно разные понятия, однако часто их путают. Для того, чтобы различать их, выведем главное различие между этими терминами. Верификация отвечает на вопрос, правильно ли создана система, а аттестация отвечает на вопрос, правильно ли работает система. Из этого следует, что верификация проверяет соответствие ПО системной спецификации, в частности функциональным и нефункциональным требованиям. Аттестация это более общий процесс. Во время аттестации цель инженера доказать заказчику, что продукт оправдывает ожидания последнего. Аттестация проводится после верификации.

На ранних этапах разработки ПО очень важна аттестация системных требований. В требованиях очень часто встречаются ошибки, недочеты, упущения, что может привести к несоответствию продукта замыслу заказчика. Инженер должен справляться с этой проблемой. Однако, как известно, сложно искоренить все погрешности в требованиях. Отдельные ошибки могут обнаружиться лишь тогда, когда программный продукт реализован.

В процессах верификации и аттестации используются две основные методики проверки и анализа систем: инспектирование ПО и тестирование ПО. Инспектирование ПО подразумевает анализ и проверку различных представлений системы, например, документации. Инспектирование происходит на всех этапах разработки программной системы. Параллельно с инспектированием может проводиться автоматический анализ исходного кода программ и соответствующих документов. Инспектирование и автоматический анализ это статические методы верификации и аттестации, поскольку им не требуется исполняемая система. Тестирование ПО есть анализ выходных данных и рабочих характеристик программного продукта для проверки правильности работы системы. Тестирование динамический метод верификации и аттестации, так как применяется к исполняемой системе.

**20.01.2020**

**Выбор критерия завершенности тестирования. Апробация работы модуля**

На практике могут быть использованы следующие критерии:

1. Время, отведенное по графика на тестирование, истекло.

2. Когда все тесты выполняются без выявления ошибок

3. На основании графика, в зависимости количества ошибок и времени их появления.

4. Если с увеличением времени тестирования, количество ошибок растет, то тестирование необходимо продолжать. Если в определенный момент наступило снижение числа выделенных ошибок и постепенно достигло малого значения, то процесс тестирования можно завершить

5. Основан на определение критерия завершения тестирования по количественным показателям надежности, которые рассчитываются по моделям надежности.

Модель надежности программного обеспечения – математическая модель, построенная для оценки надежности ПО от некоторых определенных параметров.

Необходимый шаг созревания работы — ее апробация. Верная апробация исследования — один из критериев его корректности, состоятельности, истинности итогов, один из самых реальных методов не допустить серьёзных ошибок, перекосов, преодолеть индивидуальные пристрастия исследователей, своевременно подкорректировать и поправить допущенные ошибки и недочеты.

***Апробация работы модуля:***

Апробация первого модуля тестирования прошла в рамках учебного заведения «Первого Московского образовательного комплекса» в виде презентации материала.