**09.03.2020**

**Подготовил студент: Милютин Станислав Алексеевич**

**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**

Ввиду требований к ИС было задумано создать следующие таблицы базы данных для ее функционирования:

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

**16.03.2020**

**Программирование модуля. Логическая проверка модуля**

Основная задача данного модуля в процессе практики – создать форму, на которой будут размещены:

* 2 button – войти и выйти
* 2 label – Логин и пароль
* 2 textbox – для взаимодействия с полями авторизации

Затем был написан код в формах для четкой работы полей авторизации, чтобы пользователь мог иметь свой идентификатор в сети.

**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 называют процессы проверки и анализа, в ходе которых проверяется соответствие программного обеспечения своей спецификации и требованиям заказчиков. Верификация и аттестация охватывают полный жизненный цикл ПО – они начинаются на этапе анализа требований и завершаются проверкой программного кода на этапе тестирования готовой программной системы.

Верификация и аттестация не одно и то же, хотя их легко перепутать. Кратко различие между ними можно определить следующим образом:

• верификация отвечает на вопрос, правильно ли создана система;

• аттестация отвечает на вопрос, правильно ли работает система.

Согласно этим определениям, верификация проверяет соответствие ПО системной спецификации, в частности функциональным и нефункциональным требованиям. Аттестация– более общий процесс. Во время аттестации необходимо убедиться, что программный продукт соответствует ожиданиям заказчика. Аттестация проводится после верификации, для того чтобы определить, насколько система соответствует не только спецификации, но и ожиданиям заказчика.

Как уже отмечалось ранее, на ранних этапах разработки ПО очень важна аттестация системных требований. В требованиях часто встречаются ошибки и упущения; в таких случаях конечный продукт, вероятно, не будет соответствовать ожиданиям заказчика. Но, конечно, аттестация требований не может выявить все проблемы в спецификации требований. Иногда недоработки и ошибки в требованиях обнаруживаются только после завершения реализации системы.

В процессах верификации и аттестации используются две основные методики проверки и анализа систем.

1. Инспектирование ПО. Анализ и проверка различных представлений системы, например документации спецификации требований, архитектурных схем или исходного кода программ. Инспектирование выполняется на всех этапах процесса разработки программной системы. Параллельно с инспектированием может выполняться автоматический анализ исходного кода программ и соответствующих документов. Инспектирование и автоматический анализ – это статические методы верификации и аттестации, поскольку им не требуется исполняемая система.

2. Тестирование ПО. Запуск исполняемого кода с тестовыми данными и исследование выходных данных и рабочих характеристик программного продукта для проверки правильности работы системы. Тестирование – это динамический метод верификации и аттестации, так как применяется к исполняемой системе.

На рис. 4 показано место инспектирования и тестирования в процессе разработки ПО. Стрелки указывают на те этапы процесса разработки, на которых можно применять данные методы. Согласно этой схеме, инспектирование можно выполнять на всех этапах процесса разработки системы, а тестирование – в тех случаях, когда создан прототип или исполняемая программа.

К методам инспектирования относятся: инспектирование программ, автоматический анализ исходного кода и формальная верификация. Но статические методы могут проверить только соответствие программ спецификации, с их помощью невозможно проверить правильность функционирования системы. Кроме того, статическими методами нельзя проверить такие нефункциональные характеристики, как производительность и надежность. Поэтому для оценивания нефункциональных характеристик проводится тестирование системы.

Несмотря на широкое применение инспектирования ПО, преобладающим методом верификации и аттестации все еще остается тестирование. Тестирование – это проверка работы программ с данными, подобными реальным, которые будут обрабатываться в процессе эксплуатации системы. Наличие в программе дефектов и несоответствий требованиям обнаруживается путем исследования выходных данных и выявления среди них аномальных. Тестирование выполняется на этапе реализации системы (для проверки соответствия системы ожиданиям разработчиков) и после завершения ее реализации.

На разных этапах процесса разработки ПО применяют различные виды тестирования.

1. Тестирование дефектов проводится для обнаружения несоответствий между программой и ее спецификацией, которые обусловлены ошибками или дефектами в программах. Такие тесты разрабатываются для выявления ошибок в системе, а не для имитации ее работы.

2. Статистическое тестирование оценивает производительность и надежность программ, а также работу системы в различных режимах эксплуатации. Тесты разрабатываются так, чтобы имитировать реальную работу системы с реальными входными данными. Надежность функционирования системы оценивается по количеству сбоев, отмеченных в работе программ. Производительность оценивается по результатам измерения полного времени выполнения операций и времени отклика системы при обработке тестовых данных.

Главная цель верификации и аттестации – удостовериться в том, что система "соответствует своему назначению". Соответствие программной системы своему назначению отнюдь не предполагает, что в ней совершенно не должно быть ошибок. Скорее, система должна достаточно хорошо соответствовать тем целям, для которых планировалась. Уровень необходимой достоверности соответствия зависит от назначения системы, ожиданий пользователей и условий на рынке программных продуктов.

1. Назначение ПО. Уровень достоверности соответствия зависит от того, насколько критическим является разрабатываемое программное обеспечение по тем или иным критериям. Например, уровень достоверности для систем, критическим по обеспечению безопасности, должен быть значительно выше аналогичного уровня достоверности для опытных образцов программных систем, разрабатываемых для демонстрации некоторых новых идей.

2. Ожидания пользователей. Следует с грустью отметить, что в настоящее время у большинства пользователей невысокие требования к программному обеспечению. Пользователи настолько привыкли к отказам, происходящим во время работы программ, что не удивляются этому. Они согласны терпеть сбои в работе системы, если преимущества ее использования компенсируют недостатки. Вместе с тем с начала 1990-х годов терпимость пользователей к отказам в работе программных систем постепенно снижается. В последнее время создание ненадежных систем стало практически неприемлемым, поэтому компаниям, занимающимся разработкой программных продуктов, необходимо все больше внимания уделять верификации и аттестации программного обеспечения.

3. Условия рынка программных продуктов. При оценке программной системы продавец должен знать конкурирующие системы, цену, которую покупатель согласен заплатить за систему, и назначенный срок выхода этой системы на рынок. Если у компании-разработчика несколько конкурентов, необходимо определить дату выхода системы на рынок до окончания полного тестирования и отладки, иначе первыми на рынке могут оказаться конкуренты. Если покупатели не желают приобретать ПО по высокой цене, возможно, они согласны терпеть большее количество отказов в работе системы. При определении расходов на процесс верификации и аттестации необходимо учитывать все эти факторы.

Как правило, в ходе верификации и аттестации в системе обнаруживаются ошибки. Для исправления ошибок в систему вносятся изменения. Этот процесс отладки обычно интегрирован с другими процессами верификации и аттестации. Вместе с тем тестирование (или более обобщенно – верификация и аттестация) и отладка являются разными процессами, которые имеют различные цели.

1. Верификация и аттестация – процесс обнаружения дефектов в программной системе.

2. Отладка – процесс локализации дефектов (ошибок) и их исправления (рис. 5).

Простых методов отладки программ не существует. Опытные отладчики обнаруживают ошибки путем сравнения шаблонов тестовых выходных данных с выходными данными тестируемых систем. Чтобы определить местоположение ошибки, необходимы знания о типах ошибок, шаблонах выходных данных, языке программирования и процессе программирования. Очень важны знания о процессе разработке ПО. Отладчикам известны наиболее распространенные ошибки программистов (например, связанные с пошаговым увеличением значения счетчика). Также учитываются ошибки, типичные для определенных языков программирования, например связанные с использованием указателей в языке С.

Определение местонахождения ошибок в программном коде не всегда простой процесс, поскольку ошибка необязательно находится возле того места в коде программы, где произошел сбой. Чтобы локализовать ошибки, программист-отладчик разрабатывает дополнительные программные тесты, которые помогают выявить источник ошибки в программе. Может возникнуть необходимость в ручной трассировке выполнения программы.

Интерактивные средства отладки являются частью набора средств поддержки языка, интегрированных с системой компиляции программного кода. Они обеспечивают специальную среду выполнения программ, посредством которой можно получить доступ к таблице идентификаторов, а оттуда к значениям переменных. Пользователи часто контролируют выполнение программы пошаговым способом, последовательно переходя от оператора к оператору. После выполнения каждого оператора проверяются значения переменных и выявляются возможные ошибки.

Обнаруженная в программе ошибка исправляется, после чего необходимо снова проверить программу. Для этого можно еще раз выполнить инспектирование программы или повторить предыдущее тестирование. Повторное тестирование используется для того, чтобы убедиться, что сделанные в программе изменения не внесли в систему новых ошибок, поскольку на практике высокий процент "исправления ошибок" либо не завершается полностью, либо вносит новые ошибки в программу.

В принципе во время повторного тестирования после каждого исправления необходимо еще раз запускать все тесты, однако на практике такой подход оказывается слишком дорогостоящим. Поэтому при планировании процесса тестирования определяются зависимости между частями системы и назначаются тесты для каждой части. Тогда можно трассировать программные элементы с помощью специальных контрольных примеров (контрольных данных), подобранных для этих элементов. Если результаты трассировки задокументированы, то для проверки измененного программного элемента и зависимых от него компонентов можно использовать только некоторое подмножество всего множества тестовых данных.

***Планирование верификации и аттестации***

Верификация и аттестация – дорогостоящий процесс. Для больших систем, например систем реального времени со сложными нефункциональными ограничениями, половина бюджета, выделенного на разработку системы, тратится на процесс верификации и аттестации. Поэтому очевидна необходимость тщательного планирования данного процесса.

Планирование верификации и аттестации, как один из этапов разработки программных систем, должно начинаться как можно раньше. На рис. 3 показана модель разработки ПО, учитывающая процесс планирования испытаний. Здесь планирование начинается еще на этапах создания спецификации и проектирования системы. Данную модель иногда называют V-моделью (чтобы увидеть букву V, необходимо повернуть рис. 3 на 90°). На этой схеме также показано разделение процесса верификации и аттестации на несколько этапов, причем на каждом этапе выполняются соответствующие тесты.

В процессе планирования верификации и аттестации необходимо определить соотношение между статическими и динамическими методами проверки системы, определить стандарты и процедуры инспектирования и тестирования ПО, утвердить технологическую карту проверок программ (см. раздел 19.2) и составить план тестирования программ. Чему уделить больше внимания – инспектированию или тестированию, зависит от типа разрабатываемой системы и опыта организации. Чем более критична система, тем больше внимания необходимо уделить статическим методам верификации.

В плане верификации и аттестации основное внимание уделяется стандартам процесса тестирования, а не описанию конкретных тестов. Этот план предназначен не только для руководства, он в основном предназначен специалистам, занимающимся разработкой и тестированием систем. План дает возможность техническому персоналу получить полную картину испытаний системы и в этом контексте спланировать свою работу. Кроме того, план предоставляет информацию менеджерам, отвечающим за то, чтобы у группы тестирования были все необходимые аппаратные и программные средства.

План испытаний не является неизменным документом. Его следует регулярно пересматривать, так как тестирование зависит от процесса реализации системы. Например, если реализация какой-либо части системы не завершена, то невозможно провести тестирование сборки системы. Поэтому план необходимо периодически пересматривать, чтобы сотрудников, занятых тестированием, можно было использовать на других работах.

**20.01.2020**

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

При проведении тестирования встает вопрос о том, когда завершить тестирование, когда разрабатываемое ПС достигло того уровня надежности, которое может удовлетворить будущих пользователей.

В основном на практике придерживаются следующих двух критериев: когда время, отведенное по графику на тестирование, истекло; когда все тесты выполняются без выявления ошибок.

Оба этих критерия недостаточно точны и логичны, так как первому можно удовлетворить, ничего не делая, а второй зависит от качества тестового набора данных.

Иногда используют критерий, основанный в значительной степени на здравом смысле и информации о количестве ошибок, полученных в процессе тестирования. Для этого строят график зависимости количества ошибок и времени их появления. По форме полученной кривой можно определить, стоит продолжать тестирование или нет.

На рисунке 7 приведены 2 графика, из которых видно, что если с увеличением времени тестирования число ошибок растет (левый), то тестирование необходимо про­должать. Если в процессе тестирования в определенный момент наступило снижение числа выявленных ошибок, постепенно стремится к нулю или достигло нуля, то понятно, что процесс тестирования можно завершать (правый).

*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 (рисунок 8)* – математическая модель, построенная для оценки надежности ПО от некоторых определенных параметров.

Аналитические модели дают возможность рассчитать, основываясь на данных в процессе тестирования. Аналитические модели представлены двумя группами: динамические и статические модели.

***Динамические модели:***

Поведение программного средства оценивается через появление отказов.

***Статические модели:***

Учитывают количество ошибок от количества тестовых прогонов. По области ошибок. Зависимость количества ошибок от характеристики входных данных по области.

Для использования динамических моделей. Если фиксируются интервалы каждого отказа, то получается непрерывная картина появления отказа во времени.

***Аналитические модели:***

1. Определение предположений, связанных с процедурой тестирования программного средства.

2. Разработка и выбор аналитической модели, базирующийся на предположениях о процедурах тестирования.

3. Выбор параметра модели с использованием полученных данных.

4. Применение модели. Расчет количественных показателей по модели.

***Модель Шумана:***

Исходные данные для модели Шумана собираются в процессе тестирования программного средства, в течение фиксированных или случайных временных интервалов. Каждый интервал — это стадия, на которой выполняется последовательность тестов и фиксируется число ошибок. Тестирование проводится в несколько этапов. Каждый этап представляет собой выполнение программы на полном комплексе разработанных тестовых данных. Выявленные ошибки регистрируются, но не исправляются. По завершении этапа на основе на очередном этапе тестирования может быть использована модель Шумана для расчета количественных показателей надежности и только после этого исправляются ошибки, обнаруженные на предыдущем этапе. При использовании модели Шумана предполагается, что исходное количество ошибок в программе постоянно и в процессе тестирования может уменьшаться по мере того, как ошибки выявляются и исправляются. Новые ошибки при корректировке не вносятся. Скорость обнаружения пропорционально числу ошибок. Общее число машинных инструкций в рамках одного этапа тестирования постоянно.

***Понятие апробации:***

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. Слово «апробация» латинского происхождения и означает «одобрение, утверждение, установление качеств» (не путать с опробованием, т. е. проверкой на практике).

***Формы апробации:***

Апробация может проходить в форме прилюдных отчетов, дискуссий, дискуссий, и еще в форме рецензирования (устного либо письменного) представленных работ. Официальная апробация исполненных работ нередко связана с их общественной охраной (защита плана, отчета, курсовой или же выпускной работы, диссертации). Присутствует и играет не самую последнюю роль и неофициальная апробация: разговоры, споры с экспертами и коллегами. Само собой, для апробации обязан быть представлен хотя бы в первоначальном варианте оформленный текст доклада, отчета, известия, плана, диссертации, но о процедуре литературного дизайна речь пойдет чуток ниже. Апробацию диссертации надо начинать сразу после начала работы над ней, в данном случае, кандидат сумеет получить беспристрастную оценку каждого шага проделанного им исследования, произведенных выводов и практических советов, что в нем содержатся. Проинформировать научную общественность о материалах собственной диссертации и работе над ней кандидат может, принимая участие в разных конференциях и симпозиумах, выступая на заседаниях кафедры, участвуя в разных событиях, проводимых научным обществом, кроме того у него есть возможность готовить и подталкивать во всевозможные органы собственные предложения по исследуемой теме. Помимо того, кандидат может публиковать книги, тезисы отчетов на конференциях, депонированные части собственных научных изысканий. Работу надо каждый день оговаривать с сотрудниками, преподавательским составом и научными работниками того учебного заведения, в каком готовится диссертация.

***Значение апробации:***

Конечно, апробация включает осмысление и учет образующихся вопросов, положительных и неблагоприятных оценок, отрицаний и советов. Она провоцирует доработку, более глубочайшее и аргументированное объяснение либо пересмотр ряда положений исследования и приемов подтверждения, может помочь или утвердиться в признании истинности защищаемых положений, или подкорректировать либо пересмотреть их. Мы ведем речь, по существу, об итоговой апробации исследования, хотя в общем-то может быть полезна, часто просто нужна поэтапная апробация начальных положений, гипотез, методики исследования, его переходных результатов. Положительные отзывы, в случае если они не носят комплиментарного характера, приносят удовлетворение, дают уверенность, могут помочь раскрыть возможности последующих поисков. Менее приятны, но ничуть не ниже, могут быть полезны критические замечания, рекомендации и предложения. Чрезвычайно полезным может быть глубокий анализ образовавшихся вопросов.

Есть основания признать, что могут быть полезны в конечном счете всевозможные образовавшиеся в ходе представления работы вопросы. Все связано с тем, как они воспринимаются и применяются. Из числа вопросов можно выделить уточняющие, связанные с недопониманием, неполным или же неверным осознанием изложенного. Они побуждают к уточнению, поиску более точных формулировок, доработке стиля, т.е. могут помочь сделать изложение более точным и убедительным. Вопросы, дополняющие предполагают запрос на вспомогательную информацию об источниках и перспективах становления, прецедентах, первопричинах, следствиях и т.д. Они побуждают исследователя использовать новейшие прецеденты, отдать вспомогательные оценки и мониторинги. Такие вопросы расширяют спектр оценок и раскладов. Вопросы, корректирующие побуждают к уточнению, усиливают аргументацию, уничтожают двусмысленность. Проблемные вопросы вскрывают трудности, нацеливают на более основательную интерпретацию, выводят на свежие проблемы и задачи. Стоит посоветовать чрезвычайно тщательно и дружественно воспринимать все оценки, рекомендации, советы и соратников, и оппонентов, в том числе и критические. Впрочем, воплотить стоит те из них, что могут помочь углубить изыскание, увеличить его корректность и доказательность, что не противоречит принятой концепции и вовсе не устроят работу эклектичной. Стремление же угодить всем, отреагировать на каждое замечание может усугубить или повредить изложению результатов. Апробация итогов считается одним из главных разделов введения диссертации. Оцениваются не только последствия работы, но и методика исследования, которую применил исследователь и промежуточные итоги его работы. Благодаря апробации, кандидат может переосмыслить собственные научные исследования, глубоко их подвергнуть доработке, удостовериться в надобности пересмотра каких-либо их положений. Благодаря апробации, кандидат может не только быть полноценным научным работником, но и обучиться без ошибок готовить отчеты и концерты, вести научную обсуждение вопроса, благодаря чему, у него появится возможность как следует приготовиться к защите диссертации и уверенно провести ее. Главные результаты приобретенные в диссертации представлялись соискателем в период симпозиумов и научно-практических конференций, а по ее теме публиковались учебные пособия и заметки, в каких были отражены ключевые теоретические основы работы и ее последствия, кроме того вполне возможно сказать, что последствия диссертации были интегрированы в доклад о научно-исследовательской работе.

Результаты диссертаций, имеющих прикладной и частично – теоретический нрав могут применяться во множества секторах экономики этнического хозяйства, им предоставляется возможность употребляться еще до защиты самой диссертации. Это внедрение считается введением эффектов работы и обязано отражаться в ее введении. Введение обязано быть документально утверждено той организацией, которая применила в собственной работы приобретенные соискателем последствия и возымела от их применения какой-то эффект. Эффективность внедрения результатов исследования зависит от того, как в диссертации разработаны теоретические и методические положения, а еще тем, доведены ли они до точных советов, представленных в форме нормативов, руководств и методик. Данные советы могут касаться улучшения структуры производства, нормативов временных расходов, кроме того, это могут быть аннотации по использованию всевозможных программ. Введением может считаться и внедрение результатов исследования в учебном процессе, через включение их в разные учебные материалы и пособия. Введением именуется предоставление точных итогов исследования потребителю в комфортной для него форме, способной увеличить эффективность его работы, введение постоянно обязано быть без ошибок документально оформлено.

***Апробация работы модуля:***

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