Министерство образования и науки Республики Татарстан

Государственное автономное профессиональное образовательное учреждение

«Технический колледж им. В.Д. Поташова»

Специальность: Информационные системы и программирование

Шифр: УП. 09.02.07.5771.9-5.00.00

отчет

по производственной практике

на базе автономной некоммерческой организации «Детский технопарк «Кванториум»

ПМ.01 Разработка программных модулей программного обеспечения для компьютерных систем

Разработал обучающийся IV курса Кочемасов А. А.

ФИО подпись

Руководитель практики от предприятия Сыколятин А.В.

ФИО подпись

М.П.

оценка 18.02.2023

дата

Руководитель практики Ахметлатыйпова Д.Д.

ГАПОУ «Технический колледж ФИО подпись

им. В.Д. Поташова»

оценка 18.02.2023

дата

Набережные Челны, 2023

**СОДЕРЖАНИЕ**

[1. ВВОДНЫЙ ИНСТРУКТАЖ 4](#_Toc75798735)

[1.1 Цели и задачи учебной практики 4](#_Toc75798736)

[1.2 Требования безопасности труда и пожарной безопасности 5](#_Toc75798737)

[1.3 Правила внутреннего распорядка 6](#_Toc75798738)

[1.4 Рабочее место на период практики 6](#_Toc75798739)

[1.5 Руководитель практики 6](#_Toc75798740)

[1.6 График работы на период практики 6](#_Toc75798741)

[2. Составление плана внедрения программного обеспечения 8](#_Toc75798742)

[2.1 Основные процессы и взаимосвязь между документами в информационной системе согласно стандарту ГОСТ Р ИСО/МЭК 12207. 8](#_Toc75798743)

[2.2 Виды внедрения, план внедрения. Стратегии, цели и сценарии внедрения. Типовые функции инструментария для автоматизации процесса внедрения информационной системы. 9](#_Toc75798744)

[3. Организация процесса обновления информационной системы 11](#_Toc75798745)

[3.1 Организация процесса обновления в информационной системе. Регламенты обновления. 11](#_Toc75798746)

[4. Тестирование программного обеспечения в процессе внедрения 12](#_Toc75798747)

[4.1 Тестирование программного обеспечения в процессе внедрения и эксплуатации. 12](#_Toc75798748)

[5. Составлении эксплуатационной документации 13](#_Toc75798749)

[5.1 Эксплуатационная документация. 13](#_Toc75798750)

[6. Выявление проблем совместимости программного обеспечения 14](#_Toc75798751)

[6.1 Тестирование программных продуктов. Сравнение результатов тестирования с требованиями технического задания и/или спецификацией. Анализ рисков. Выявление первичных и вторичных ошибок. 14](#_Toc75798752)

[7. Выявление причин возникновения проблем совместимости программного обеспечения 16](#_Toc75798753)

[7.1 Выполнение чистой загрузки. Выявление причин возникновения проблем совместимости программного обеспечения. Выбор методов выявления совместимости. 16](#_Toc75798754)

[7.2 Проблемы перехода на новые версии программ. Мастер совместимости программ. Инструментарий учета аппаратных компонентов 18](#_Toc75798755)

[8. Использование динамически загружаемых библиотек 20](#_Toc75798756)

[8.1 Устранение проблем совместимости программного обеспечения. 20](#_Toc75798757)

[8.2 Конфигурирование программных и аппаратных средств. 20](#_Toc75798758)

[9. Решение проблем конфигурации с помощью групповых политик 22](#_Toc75798759)

[9.1 Групповые политики. Аутентификация. Учетные записи. 22](#_Toc75798760)

[10. Тестирование на совместимость в безопасном режиме 23](#_Toc75798761)

[10.1 Понятие совместимости программного обеспечения. Аппаратная и программная совместимость. Совместимость драйверов. Причины возникновения проблем совместимости. Методы выявления проблем совместимости программного обеспечения 23](#_Toc75798762)

[11. Cоздание в системе виртуальной машины для исполнения приложений 26](#_Toc75798763)

[11.1 Создание в системе виртуальной машины для исполнения приложений. Изменение настроек по умолчанию в образе. Подключение к сетевому ресурсу. Настройка обновлений программ. Обновление драйверов 26](#_Toc75798764)

[12. Создание образа системы 28](#_Toc75798765)

[12.1 Создание образа системы. Восстановление системы. Разработка модулей программного средства. Настройка сетевого доступа. 28](#_Toc75798766)

[13. Настройка обновлений программ 29](#_Toc75798767)

[13.1 Установка и настройка антивируса. Настройка обновлений с помощью зеркала 29](#_Toc75798768)

[14. Оптимизация использования памяти и жесткого диска 30](#_Toc75798769)

[14.1 Оптимизация использования памяти. Оптимизация использования жесткого диска. Оптимизация использования сети. Инструменты повышения производительности программного обеспечения. 30](#_Toc75798770)

[ЗАКЛЮЧЕНИЕ 34](#_Toc75798771)

[СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ И ЛИТЕРАТУРЫ 35](#_Toc75798772)

1. ВВОДНЫЙ ИНСТРУКТАЖ
   1. Цели и задачи учебной практики

Задачей производственной практики по специальности 09.02.07 Информационные системы и программирование является освоение вида профессиональной деятельности: Осуществление интеграции программных модулей, т. е. систематизация, обобщение, закрепление и углубление знаний и умений, формирование общих и профессиональных компетенций, приобретение практического опыта в рамках профессионального модуля Осуществление интеграции программных модулей, предусмотренного ФГОС СПО.

С целью овладения видом профессиональной деятельности обучающийся в ходе практики должен:

**иметь практический опыт:**

* интеграции модулей в программное обеспечение;
* отладке программных модулей.

**уметь:**

* использовать выбранную систему контроля версий;
* использовать методы для получения кода с заданной функциональностью и степенью качества.

**знать:**

* модели процесса разработки программного обеспечения;
* основные принципы процесса разработки программного обеспечения;
* основные подходы к интегрированию программных модулей;
* основы верификации и аттестации программного обеспечения**.**

Результатом производственной практики является освоение общих (ОК) компетенций:

|  |  |
| --- | --- |
| **Код** | **Наименование общих компетенций** |
| ОК 1 | Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам |
| ОК 2 | Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности. |
| ОК 3 | Планировать и реализовывать собственное профессиональное и личностное развитие. |
| ОК 4 | Работать в коллективе и команде, эффективно взаимодействовать с коллегами, руководством, клиентами. |
| ОК 5 | Осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста. |
| ОК 6 | Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных общечеловеческих ценностей |
| ОК 7 | Содействовать сохранению окружающей среды, ресурсосбережению, эффективно действовать в чрезвычайных ситуациях. |
| ОК 8 | Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности |
| ОК 9 | Использовать информационные технологии в профессиональной деятельности. |
| ОК 10 | Пользоваться профессиональной документацией на государственном и иностранном языках |
| ОК 11 | Планировать предпринимательскую деятельность в профессиональной сфере |

и профессиональных (ПК) компетенций:

|  |  |
| --- | --- |
| **Код** | **Наименование видов деятельности и профессиональных компетенций** |
| ВД 2 | Осуществление интеграции программных модулей |
| ПК 2.1. | Разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент |
| ПК 2.2. | Выполнять интеграцию модулей в программное обеспечение |
| ПК 2.3 | Выполнять отладку программного модуля с использованием специализированных программных средств |
| ПК 2.4 | Осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения. |
| ПК 2.5. | Производить инспектирование компонент программного обеспечения на предмет соответствия стандартам кодирования |

* 1. Требования безопасности труда и пожарной безопасности

Настоящая инструкция по охране труда программиста, занятого эксплуатацией персональных электронно-вычислительных машин (ПЭВМ) и видео дисплейных терминалов (ВДТ), разработана с учетом условий его работы в конкретной организации.

На программиста могут воздействовать опасные и вредные производственные факторы:

физические:

* повышенные уровни электромагнитного излучения;
* повышенные уровни рентгеновского излучения;
* повышенные уровни ультрафиолетового излучения;
* повышенный уровень инфракрасного излучения;
* повышенный уровень статического электричества;
* повышенные уровни запыленности воздуха рабочей зоны;
* повышенное содержание положительных аэроионов в воздухе рабочей зоны;
* пониженное содержание отрицательных аэроионов в воздухе рабочей зоны;
* пониженная или повышенная влажность воздуха рабочей зоны;
* пониженная или повышенная подвижность воздуха рабочей зоны;
* повышенный уровень шума;
* повышенный или пониженный уровень освещенности;
* неравномерность распределения яркости в поле зрения;
* повышенная яркость светового изображения;
* повышенный уровень пульсации светового потока;
* повышенное значение напряжения в электрической цепи, замыкание которой может произойти через тело человека;
* химические:
* повышенное содержание в воздухе рабочей зоны двуокиси углерода, озона, аммиака, фенола, формальдегида и полихлорированных бифенилов;
* психофизиологические:
* напряжение зрения;
* напряжение внимания;
* эмоциональные нагрузки;
* длительные статические нагрузки;
* монотонность труда;
* большой объем информации, обрабатываемой в единицу времени;
* нерациональная организация рабочего места;
* биологические:
* повышенное содержание в воздухе рабочей зоны микроорганизмов.
* К работам программистом допускаются:
* лица не моложе 18 лет, прошедшие обязательный при приеме на работу и ежегодные медицинские освидетельствования на предмет пригодности для работы;
* прошедшие вводный инструктаж по охране труда;
* прошедшие обучение безопасным приемам и методам труда по программе, утвержденной руководителем предприятия (работодателем), разработанной на основе Типовой программы, и прошедшие проверку знаний, в том числе по электробезопасности;
* прошедшие курс обучения на персональном компьютере с использованием конкретного программного обеспечения;
* прошедшие инструктаж по охране труда на конкретном рабочем месте по данной инструкции.

Программист должен быть обеспечен СИЗ в соответствии с Межотраслевыми правилами обеспечения работников специальной одеждой, специальной обувью и другими средствами индивидуальной защиты, утвержденными Приказом Минздравсоцразвития России от 01.06.2009 N 290н; выдаваемые работникам средства индивидуальной защиты должны соответствовать характеру и условиям работы и обеспечивать безопасность труда.

Характеристика выданных СИЗ (номенклатура, срок выдачи и нормы соответствия) устанавливается из личных карточек работников, занятых на определенном рабочем месте. Нормативные номенклатура и сроки выдачи СИЗ определяются согласно Типовым отраслевым нормам бесплатной выдачи рабочим и служащим специальной одежды, специальной обуви и других СИЗ.

* 1. Правила внутреннего распорядка

Студент должен явиться на производственную практику в точно установленное время и соблюдать правила пожарной безопасности. Рабочий день длится 6 часов.

На производственную практику обучающийся должен явиться в аккуратно подогнанной, чистой одежде.

Во время производственной практики предусмотрены 10-ти минутные перерывы для отдыха -через каждые 50 минут.

Студент должен выходить на перерыв по звонку, с разрешения руководителя и вовремя начинать работу после перерыва.

Во время производственной практики студент должен придерживаться установленного порядка. Не разрешается без дела ходить по кабинету, мешать товарищам в работе, нарушать нормы и правила охраны труда, бесцельно включать и выключать свет, пускать и останавливать персональные компьютеры.

Обучающийся обязан бережно относиться к государственному имуществу, имуществу предприятия (предметам мебели, компьютерам и т.д.). Порча имущества влечет за собой дисциплинарную и материальную ответственность.

Обучающийся обязан в течение рабочего дня поддерживать порядок на рабочем месте, после работы тщательно произвести уборку рабочего места.

* 1. Рабочее место на период практики

Рабочее место на период практики – ООО «АЗС-Строй», ул. Советская, д. 178, г.Альметьевск, Республика Татарстан.

* 1. Руководитель практики

Руководители практики от образовательного учреждения – Ахметлатыйпова Д.Д.

Руководители практики от предприятия – Буднев А.В.

* 1. График работы на период практики

|  |  |
| --- | --- |
| День недели | Время |
| Понедельник | 8:00-14:00 |
| Вторник | 8:00-14:00 |
| Среда | 8:00-14:00 |
| Четверг | 8:00-14:00 |
| Пятница | 8:00-14:00 |
| Суббота | выходной |
| Воскресенье | выходной |

1. Составление плана внедрения программного обеспечения
   1. Основные процессы и взаимосвязь между документами в информационной системе согласно стандарту ГОСТ Р ИСО/МЭК 12207.

Одно из центральных мест здесь занимает стандарт ГОСТ Р ИСО/МЭК 12207 "Процессы жизненного цикла программных систем" (ГОСТ 12207, 1999) - один из самых известных и распространенных процессно-ориентированных стандартов в области управления ИТ. Ссылки на него встречаются практически во всех работах и методиках, относящихся к процессам управления ИТ.

Область применения его, как следует из названия, относительно узка: процессы, выполняющиеся в ходе жизненного цикла программной системы (речь не идет о жизненном цикле технических средств, таких как вычислительные мощности, сети передачи данных и т. п.). Эти процессы представлены во взаимосвязи с другими процессами организации. Модель жизненного цикла стандарт определяет как "структуру, состоящую из процессов, работ и задач, включающих в себя разработку, эксплуатацию и сопровождение программного продукта, охватывающую жизнь системы от установления требований к ней до прекращения ее использования".

Методологическая основа ГОСТ Р ИСО/МЭК 12207 - разбиение процессов на группы, которых в стандарте вводится три.

* Основные. Это процессы, непосредственно относящиеся к жизненному циклу информационной системы. Можно считать, что это производственные процессы организации.
* Вспомогательные. Это процессы, предназначенные для поддержки основных процессов. Сами по себе эти процессы организации не нужны - только в связи с основными процессами, которые они обслуживают. Несколько процессов из этой группы связано с управлением качеством.
* Организационные. Это общекорпоративные процессы, такие как "Обучение" или "Управление". Эти процессы существуют в организации независимо от того, как организовано производство и как устроены вспомогательные процессы.

Если на протяжении жизненного цикла программной системы возникает задача, которая имеет общекорпоративный характер, например задача обучения, используется соответствующий процесс из стандарта, а не общекорпоративный процесс. Возможно, это решение было принято просто для того, чтобы формально обеспечить замкнутость и, следовательно, самодостаточность стандарта. Как следствие, организационные процессы должны быть как-то увязаны с корпоративными стандартами организации.

Процессы, согласно стандарту, состоят из работ, работы - из задач. Последовательность работ и задач, приведенная в стандарте, не является жесткой. Необходимо только выдерживать логические связи между работами и задачами.

Цель стандарта - определить полную совокупность процессов, которые могут выполняться в ходе проекта по созданию программной системы. Но поскольку проекты могут сильно различаться, например по масштабам, сложности, рискам и т. п., допускается для каждого проекта локально видоизменять использующиеся в нем процессы, исключая или добавляя отдельные работы и задачи. Такая деятельность называется в стандарте адаптацией. Стандарт содержит описание процесса собственной адаптации. Вот фрагмент обязательного Приложения А "Процесс адаптации", описывающий одну из работ (она имеет номер А.3) процесса адаптации.

* начать создавать общекорпоративный процесс обучения параллельно с проектом автоматизации;
* создать локальный процесс обучения в ходе проекта, т. е. включить разработку процесса в договор;
* исключить процесс обучения в ходе адаптации стандарта.

Какой вариант выбрать - дело организации, но сам выбор сделать придется, и одно из привлекательных свойств ГОСТ Р ИСО/МЭК 12207 состоит как раз в том, что он явно говорит о необходимости такого выбора. Как следствие, при планировании проекта автоматизации необходимо будет запланировать соответствующие работы.

Рассмотрен ГОСТ Р ИСО/МЭК 12207 - один из наиболее удачных и широко известных процессных стандартов последнего времени. Проанализированы новые идеи, впервые появившиеся в этом стандарте, и его ограничения. Особо отмечается сложность организации внедрения стандарта.

* 1. Виды внедрения, план внедрения. Стратегии, цели и сценарии внедрения. Типовые функции инструментария для автоматизации процесса внедрения информационной системы.

После сформировавшейся мысли о том, что компании необходимо внедрение информационной системы управления предприятием, нужно определиться, кто этим будет заниматься. Существует несколько подходов к запуску проектов такого рода на предприятии:

* заключение контракта с крупной компанией, внедряющей ИС. К преимуществам можно отнести опыт компании-аутсорсера и отдельных ее специалистов, а также наличие собственных проектных наработок. К недостаткам относят стоимость работ, возможную текучку кадров и возможность того, что за громким именем могут стоять не самые лучшие специалисты;
* приглашение небольшой, региональной IT-компании. Однозначным плюсом является высокая вероятность того, что внедрение автоматизированной информационной системы станет приоритетным проектом для нее. Если проект предстоит крупный, а значит долгий, стоит опасаться внезапных смен руководства, специалистов и приоритетов небольших фирм-внедренцев;
* внедрение силами собственного IT-отдела. В этом варианте привлекает отсутствие дополнительных трат, постоянная связь со специалистами и возможность лично управлять проектом. Однако тут кроется и большая опасность – специалисты IT-отдела, зачастую зависящие от пользователей и руководства, полностью ориентируются их решения, в том числе не всегда правильные;
* приглашение эксперта. Отличный способ сэкономить и получить специалиста в нужной области. К недостаткам можно отнести необходимость высокой организованности всех сотрудников компании, зависимость успеха от одного человека и формальную ответственность за проект.

Практика показывает, что управление внедрением информационных систем лучше доверить опытным специалистам. Именно поэтому, какой бы вариант команды внедрения вы не выбрали, обязательно проверяйте опыт – и не только количественный, но и качественный. Проверяйте отзывы о работе IT-компаний и экспертов, следите за квалификацией собственных специалистов.

Такой важный момент, как стратегический план внедрения и выбор типа взаимоотношений с внедренцами информационной системы, важен, но не является единственным критерием. Эффективность внедрения информационной системы на предприятии зависит от нескольких факторов и готовности персонала учитывать их в работе. Специалисты выделяют несколько основных правил, игнорирование которых с большой вероятностью приведет к печальным последствиям:

* осознание необходимости внедрять современные технологические инструменты и готовность к внедрению всех сотрудников;
* изучение основ построения системы;
* грамотный выбор подходящей системообразующей программы и команды, отвечающей за ее внедрение;
* выделение квалифицированных кадров для контроля проекта со стороны заказчика;
* последовательная и четкая организация проекта;
* желание меняться к лучшему.

Сложно определить какие-либо временные рамки запуска корпоративной информационной системы. Многое будет зависеть от того, существовала ли раньше разработка информационных систем предприятия или придется начинать с нуля. Следует приготовиться к тому, что внедрение займет достаточно много времени и существенно изменит некоторые бизнес-процессы в компании. Технология внедрения информационных систем может существенно отличаться у разных специалистов, но определенные этапы выделяются практически в каждом успешном проекте.

Успешные внедрения информационных систем включают в себя достаточно много важных и полезных для предприятия этапов. Помимо непосредственно начала работы в ИС, они помогают компании упорядочить ключевые бизнес-процессы и выявить проблемные места. Чем крупнее компания, тем больше принято сотрудников, чьи функции и квалификация вызывают вопросы. Грамотное внедрение информационной системы выявит их.

Если компания хочет не просто «для галочки» внедрить ИС, а действительно эффективно пользоваться всеми ее возможностями, предстоят следующие этапы:

* В первую очередь необходимо определить цель внедрения. Многие руководители высшего звена поверхностно относятся к этому этапу, но на самом деле он задает направление всему внедрению ИС;
* Обследование бизнес-процессов компании. В этот этап входят интервью с менеджментом, рядовыми сотрудниками, составление схем по каждому процессу. На выходе получается уточнение целей внедрения и возможность предварительно оценить объем работ и стоимость;
* Составление проекта, технического задания и регламента. В этих документах должны быть описаны все бизнес-процессы, участвующие во внедрении ИС. Старайтесь составлять проект внедрения максимально подробно, с указанием необходимых данных, их структуры, алгоритмов действий, рабочих мест;
* Подготовка специалистов. Сотрудники компании при начале внедрения должны знать, что от них требуется, чтобы не задерживать выполнение работы. Также администраторы и разработчики компании должны начать разбираться в информационной системе. То есть сотрудники расширяют свои знания на благо компании;
* Настройка информационной системы в соответствии со спецификой предприятия. В этот этап включается:
* Разграничение прав на функционал системы для сотрудников;
* Начальное заполнение данных;
* Настройка алгоритмов расчетов, создание необходимых отчетов.
* Тестирование информационной системы. На этом этапе могут обнаружиться проблемы внедрения в разрезе алгоритмов или необходимость в новых отчетах;
* Опытная эксплуатация с реальными данными. Чаще всего на этом этапе многие сотрудники компании выполняют больше работы. Им приходится не только работать, как раньше, но и отражать свои действия в информационной системе. Требуется максимальная дисциплина и сосредоточение усилий всех участников внедрения. Конечным результатом должно стать совпадение данных информационной системы с реальным положением дел;
* Промышленная эксплуатация. На этом этапе осуществляется переход сотрудников на полноценную работу в информационной системе. Должна быть организована техническая поддержка пользователей;
* Завершение проекта. Основным результатом этапа являются подписанные должностные инструкции, разграничение обязанностей подразделений и их взаимодействия. Корпоративная информационная система запущена на предприятии.

1. Организация процесса обновления информационной системы
   1. Организация процесса обновления в информационной системе. Регламенты обновления.

**Обновления** представляют собой дополнения к программному обеспечению, предназначенные для предотвращения или устранения проблем и улучшения работы компьютера. Обновления безопасности для Windows способствуют защите от новых и существующих угроз для конфиденциальности и устойчивой работы компьютера. Оптимальный способ получения обновлений безопасности - включить автоматическое обновление Windows и всегда оставаться в курсе последних проблем, связанных с безопасностью и предоставить операционной системе самостоятельно заботиться о своей безопасности. В этой статье речь пойдет именно о Центре обновления Windows.

Желательно обновлять компьютер как можно чаще. В этом случае использования автоматического обновления, операционная система Windows устанавливает новые обновления, как только они становятся доступными. Если не устанавливать обновления, то компьютер может подвергнуться риску в плане безопасности или же могут возникнуть нежелательные неполадки в работе Windows или программ.

Каждый день появляется все больше и больше новых вредоносных программ, использующих уязвимости Windows и другого программного обеспечения для нанесения ущерба и получения доступа к компьютеру и данным. Обновления Windows и другого программного обеспечения позволяют устранить уязвимости вскоре после их обнаружения. Если отложить установку обновлений, компьютер может стать уязвимым для таких угроз.

Обновления и программное обеспечение от Microsoft для продуктов Microsoft являются бесплатным предложением от службы поддержки, так что можно не волноваться за то, что с вас будет взиматься дополнительная плата за обеспечение надежности вашей системы. Чтобы узнать, являются ли обновления других программ бесплатными, обращайтесь к соответствующему издателю или изготовителю. При загрузке и установке обновлений различных программ в зависимости от типа подключения к Интернету может взиматься стандартная плата за местные или междугородные телефонные переговоры, а также плата за пользование Интернетом. В связи с тем, что обновления применяются к Windows и установленным на компьютере программам независимо от того, кто ими пользуется, после установки обновлений они будут доступны для всех пользователей компьютера.

Все обновления подразделяются на

* Важные обновления обеспечивают существенные преимущества в безопасности, конфиденциальности и надежности. Их следует устанавливать сразу же, как только они становятся доступны, и можно выполнять установку автоматически с помощью «Центра обновления Windows».
* Рекомендуемые обновления могут устранять менее существенные проблемы или делать использование компьютера более удобным. Хотя эти обновления не предназначены для устранения существенных недостатков в работе компьютера или программного обеспечения Windows, их установка может привести к заметным улучшениям. Их можно устанавливать автоматически.

1. Тестирование программного обеспечения в процессе внедрения
   1. Тестирование программного обеспечения в процессе внедрения и эксплуатации.

Любая программа может содержать в себе ошибки. Компилятор способен выявлять только синтаксические ошибки, но не способен отслеживать семантику. Большинство ошибок проявляется в ходе работы программы, при этом они могут возникать не всегда, а лишь при определенных условиях. Таким образом, успешная компиляция программы и выполнение этой программы в одних и тех же условиях не гарантируют отсутствие ошибок.

Для выявления ошибок в программах ЖЦ разработки ПО предусматривает процесс тестирования, который является достаточно трудоемким и занимает больше времени, чем кодирование. (Г. Майерс дает оценку 1/3 для тестирования, при том, что кодированиезанимает примерно 1/6.) Тестируемое ПО обычно называют SUT - Software Under Test. Цель тестирования - не убедиться в безошибочной работоспособности программы, а наоборот - найти ошибки. Поэтому в первую очередь возникает вопрос: а что есть ошибка в программе?

Заметим, что к этому моменту программа уже представляет собой выполнимый процессором набор команд, т.е. с точки зрения процессора она корректна. Даже если при каких-то условиях программа аварийно завершает свое выполнение или "портит" другие процессы, сразу нельзя сказать, что это ошибка в программе - возможно, так было задумано. Таким образом, ошибки необходимо рассматривать с точки зрения пользователя, основываясь на дополнительной информации, т.е. неком описании того, что должна делать программа (это же описание может включать в себя требование о том, чтобы программа никогда не завершалась аварийно и др.). При рассмотрении вопросов анализа программного кода порой удобнее применять ранее рассмотренную в разделе 1 модель жизненного цикла. Ее часто называют V-образной из-за расположения блоков на рисунке (рис. 1).

Нисходящая левая ветвь модели отражает поэтапную последовательность преобразования одних программных документов в другие: SYS - системных требований в SRD - требования к программному обеспечению, проектированию и формированию DDD - описания архитектуры системы и, наконец, разработке CODE - кода программ. Восходящая правая ветвь отражает процесс верификации разработанного программного обеспечения.

На первом этапе путем тестирования производится модульная верификация (MV), при которой поведение исполняемого программного кода проверяется на соответствие его DDD-описанию. Это наиболее трудоемкая и скрупулезная часть исследования. Она часто требует написания драйверов - моделей модулей, вызывающих процедуры тестируемого модуля, и заглушек - моделей процедур других модулей, вызываемых из тестируемого. Часто в MV отдельно выделяют процесс тестирования межмодульных связей, описанных в DDD.

На втором этапе производится комплексная верификация (CV) реализованного программного обеспечения по отношению к требованиям. Наконец, производится комплексная интеграция (CI) и проверка всей системы: пользователь, аппаратура и программное обеспечение. При грамотном процессе разработки уже на этапах нисходящей ветви для каждого требования определяется, на каком уровне верификации должна будет проводиться проверка его соблюдения.

При этом следует исходить из предположения, что ошибки всегда есть. Тестирование можно считать успешным, если найдены ошибки, а не наоборот. В достаточно сложном ПО все ошибки могут не обнаруживаться даже после длительного тестирования, однако, чем тщательнее ведется тестирование, тем меньше ошибок остается и тем менее вероятно возникновение невыявленных ошибок.

1. Составлении эксплуатационной документации
   1. Эксплуатационная документация.

Эксплуатационная документация – это вид технической документации, описывающий порядок установки, настройки и использования разработанного программного обеспечения (автоматизированной системы).

Как правило, документы данного вида являются составляют либо дополнение к "ядру" комплекта технической документации (проектной документации), либо полностью независимы и автономны, а в некоторых случаях являются единственной частью документации на программный продукт. Именно эксплуатационную документацию мы чаще всего подразумеваем, когда говорим о программной документации в целом.

Основные документы, входящие в состав эксплуатационной документации:

* руководство пользователя
* руководство администратора
* руководство программиста

Стоит отметить, что все эти документы – это лишь основной перечень в соответствии с ролями пользователей (программист, администратор, обычный пользователь). Однако в комплекте эксплуатационной документации помимо Руководства пользователя очень часто встречается Краткое руководство пользователя, Руководство по быстрому старту и т.д. – все эти документы по сути также являются Руководством пользователя, но представлены в несколько иных вариациях.

Отечественные стандарты, разрабатывающие эксплуатационную документацию:

* **ЕСКД**, Единая система конструкторской документации (ГОСТ 2);
* **ЕСПД**, Единая система программной документации (ГОСТ 19);
* **КСАС**, Комплекс стандартов на автоматизированные системы (ГОСТ 34).

Каждая из вышеперечисленных серий ГОСТ предъявляет свои требования к составу, порядку разработки и ведения комплектов эксплуатационной документации, а также к содержанию и оформлению отдельных документов комплекта.

1. Выявление проблем совместимости программного обеспечения
   1. Тестирование программных продуктов. Сравнение результатов тестирования с требованиями технического задания и/или спецификацией. Анализ рисков. Выявление первичных и вторичных ошибок.

Тестирование является одним из наиболее устоявшихся способов обеспечения качества разработки программного обеспечения и входит в набор эффективных средств современной системы обеспечения качества программного продукта.

Качество программного продукта характеризуется набором свойств, определяющих, насколько продукт "хорош" с точки зрения заинтересованных сторон, таких как заказчик продукта, спонсор, конечный пользователь, разработчики и тестировщики продукта, инженеры поддержки, сотрудники отделов маркетинга, обучения и продаж. Каждый из участников может иметь различное представление о продукте и о том, насколько он хорош или плох, то есть о том, насколько высоко качество продукта. Таким образом, постановка задачи обеспечения качества продукта выливается в задачу определения заинтересованных лиц, их критериев качества и затем нахождения оптимального решения, удовлетворяющего этим критериям.

Тестирование программного продукта должно проводиться прям с самого начала его создания. Но при этом, сами опытные разработчики в тестировании не должны принимать участия, так как не царское это дело. Тестировать программный продукт должны специально обученные сотрудники, называемые тестировщиками, ибо даже самый опытный разработчик не сможет увидеть свою ошибку, даже с использованием самых новейших оптических приборов.

Тем не менее, все разработчики сходятся во мнении, что тестирование программного продукта с точки зрения классификации по целям должно делиться на два класса:

* Функциональное тестирование
* Нефункциональное тестирование

Под функциональным тестированием понимается проверка соответствия программного продукта функциональным требованиям, указанным в техническом задании на создание это продукта. Если говорить проще, то при функциональном тестировании проверяется выполняет ли программный продукт все функции, которые должен.

Для проведения функционального тестирования персоналом отдела технического контроля разрабатывается документ программа и методика испытаний функционала приложения (ПМИ). Документ ПМИ содержит перечень сценариев тестирования программного продукта (test cases) с подробным описанием шагов. Каждый шаг сценария тестирования характеризуется действиями пользователя (специалиста по тестированию) и ожидаемыми результатами – ответной реакции программы на эти действия. Программа и методика испытаний обязана имитировать эксплуатацию программного продукта в реальном режиме. Это означает, что сценарий тестирования должен быть построен на основе анализа операций, которые будут выполнять будущие пользователи системы, а не быть искусственно составленной последовательностью понятных только разработчику манипуляций.

Обычно, функциональное тестирование проводится на двух уровнях:

* Компонентное (модульное) тестирование. Тестирование отдельных компонентов программного продукта, сфокусированное на их специфике, назначении и функциональных особенностях.
* Интеграционное тестирование. Данный вид тестирования проводится после компонентного тестирования и направлен на выявление дефектов взаимодействия различных подсистем на уровне потоков управления и обмена данными.

Нефункциональное тестирование оценивает такие качества программного продукта, как, например, эргономику или производительность.

Думаю, важность данного вида тестирования понятна и не требует обоснования. Ведь всем понятно, что если, к примеру, производительность системы не достаточна, то пользователям придется по пол дня ждать отклика на свои действия, что может привести к их массовой спячке.

Как следует из названия, при нефункциональном тестировании проверяется соответствие программного продукта нефункциональным требованиям из технического задания на его создание. И, как в случае с функциональным тестированием, для нефункционально разрабатывается программа и методика испытаний.

1. Выявление причин возникновения проблем совместимости программного обеспечения
   1. Выполнение чистой загрузки. Выявление причин возникновения проблем совместимости программного обеспечения. Выбор методов выявления совместимости.

Чистая загрузка в Windows 10, 8 и Windows 7 (не путать с чистой установкой, под которой понимается установка ОС с флешки или диска с удалением предыдущей системы) позволяет исправить проблемы с системой, вызванные неправильной работой программ, конфликтами ПО, драйверов и служб Windows.

В чем-то чистая загрузка (clean boot) схожа с безопасным режимом, но не является тем же самым. В случае входа в безопасном режиме, в Windows отключается почти всё, что не является обязательным для запуска, а для работы используются «стандартные драйвера» без аппаратного ускорения и других функций (что может быть полезно при исправлении проблем с оборудованием и драйверами).

При использовании чистой загрузки Windows, предполагается, что с сама операционная система и оборудование работают исправно, а при запуске не загружаются компоненты от сторонних разработчиков. Данный вариант запуска подходит для тех случаев, когда нужно выявить именно проблемное или конфликтующее программное обеспечение, сторонние службы, мешающие нормальной работе ОС. Важно: для того, чтобы настроить чистую загрузку, вы должны быть администратором в системе.

Т.е. действия по включению чистой загрузки будут следующими:

* Нажать Win+R, ввести *msconfig*, нажать «Ок».
* На вкладке «Общие» выбрать «Выборочный запуск» и снять отметку с «Загружать элементы автозагрузки».
* На вкладке «Службы» включить «Не отображать службы Майкрософт», а затем выключить все сторонние службы.
* Нажать Ок и перезагрузить компьютер.

Программные неисправности компьютера встречаются намного чаще, чем аппаратные. Конечно, ничего хорошего в их появлении нет, однако радует то, что такие неисправности можно без труда и риска исправить в домашних условиях.

Рассмотрим основные причины возникновения подобного рода ошибок.

**Несовершенство программного обеспечения**. Не все люди, которые занимаются написанием программного обеспечения, являются высококвалифицированными программистами. В основном это самоучки, начитавшиеся умных книг, знакомые с основами программирования. Написав программу, которая работает у него на компьютере, такой «программист» решает, что она будет работать и у других, совершенно не учитывая тот факт, что на компьютерах пользователей могут быть установлены разные операционные системы, компьютеры могут обладать разными ресурсами, устройствами, использовать разное программное обеспечение и т. д.

Это, конечно, совсем не означает, что профессионалы пишут программы без проблем: их преследуют те же ошибки. Однако квалифицированные программисты умеют адаптировать программу для работы в разных условиях, что делает ее менее уязвимой и, соответственно, более корректной в работе.

Очень часто бывает так, что человек садится за написание программы, не думая о том, что при работе она будет захватывать различные ресурсы, контролировать которые в одиночку не удастся. В результате появляется программное обеспечение, имеющее множество недочетов и постоянно выдающее ошибки.

Достаточно большой проблемой является и то, как программа распоряжается установленными в операционной системе системными библиотеками. Иногда системные библиотеки заменяются улучшенными версиями, с которыми отказываются работать другие программы, что, в свою очередь, также приводит к появлению ошибок.

**Несовершенство операционной системы**. Какими бы совершенными ни были операционные системы, они никогда не смогут создать нормальные условия для работы всего существующего программного обеспечения. Кроме того, совместимость операционных систем с выпуском каждой новой системы ухудшается. В таких условиях разработчики вынуждены писать программы, ориентированные на конкретную операционную систему.

Пользователю остается или работать с сертифицированными программами, или мириться со сложившейся ситуацией. Иногда и выбирать не приходится, ведь многие программы распространяются бесплатно (можно догадаться, какое у большой части из них качество работы).

**Отсутствие нужных ресурсов.** Часто даже небольшая программа требует для нормальной работы большее количество ресурсов, чем ей может предоставить операционная система. В таком случае для обеспечения работы программы система увеличивает файл подкачки, отбирающий оперативную память у других программ. Это приводит не только к снижению производительности компьютера, но и к появлению разнообразных ошибок.

Обычно такую проблему можно решить простым увеличением объема оперативной памяти. Однако это можно сделать и с помощью специальных утилит.

**Ошибки в реестре.** Реестр — мозг операционной системы. Ошибки в нем негативно сказываются на всех процессах, происходящих в компьютере. Причиной возникновения сбоев в реестре являются все те же непрофессионально написанные программы, прописывающие свои файлы и ссылки в самых различных местах. Не стоит также забывать о троянских конях и «червях», которые только и мечтают о том, чтобы что-нибудь подпортить.

Для «лечения» реестра предназначены специальные утилиты, умеющие анализировать записи и удалять ошибочные и не используемые данные. При этом не следует забывать об элементарном сохранении рабочей версии файлов реестра.

**Конфликты между устройствами**. Аппаратное обеспечение, как и программное, также не всегда бывает полностью совместимым. При использовании устройств доморощенных производителей вы обязательно получите целый букет подобных конфликтов. Например, перестанет работать звуковая карта или пропадет один из приводов компакт-дисков.

Прежде всего в такой ситуации следует проанализировать используемые конфликтующими устройствами прерывания и ресурсы. Для этого воспользуйтесь Диспетчером устройств или просмотрите отчеты об ошибках работы системы.

Достаточно часто в инструкции к устройству производители указывают несовместимые с ним комплектующие других производителей. Поэтому при сборке или модернизации компьютера обязательно внимательно изучите документацию к каждому из приобретенных устройств.

Часто подобные конфликты можно решить переустановкой платы расширения в другой слот или вручную, назначив им свободные ресурсы (с помощью Диспетчера устройств в свойствах конкретного устройства можно назначить прерывания и диапазон адресов). В случае с Windows Vista еще один действенный способ — замена проблемного устройства более новой моделью.

**Вирусы, троянские кони и «черви».** Если вы пользуетесь Интернетом, то вам не избежать неприятных сюрпризов, таких как различные вирусы, трояны, «черви», программы-шпионы и т. д.

Трудно сказать, когда Интернет наполнился вирусами, однако на сегодняшний день ни один пользователь Интернета не в состоянии полностью обезопасить свой компьютер от этой напасти. По данным некоторых исследований, на компьютерах, часто работающих с Интернетом, может находиться одновременно до 30 различных модулей описанного характера. Порой разного рода троянские кони и «черви» проникают даже сквозь, казалось бы, непробиваемую защиту антивирусной программы и брандмауэра.

Однако в любом случае необходимо использовать хорошую антивирусную программу, а также программу поиска модулей-шпионов и блокирования программных портов (брандмауэр). В противном случае вы обречете себя на частую переустановку операционной системы.

**Ограничения операционной системы**. Чтобы обезопасить свои системные файлы и вашу работу за компьютером в целом, Windows Vista часто не разрешает выполнять те или иные действия: устанавливать новые программы, удалять файлы, просматривать веб-ресурсы и т. д.

Уменьшение [проблем с совместимостью](https://datbaze.ru/article/ispravlenie-nepoladok-sovmestimosti-shims.html) приложения обычно зависит от различных факторов, таких как тип приложения и текущей поддержки приложения. Некоторые из общих методов включают следующее:

* Изменение конфигурации существующего приложения: Вы можете использовать инструменты, Compatibility Administrator или Standard User Analyzer (устанавливается с ACT), для обнаружения проблемы и создания исправления данного приложения, что решит проблему совместимости.
* Применение обновлений или пакетов обновлений к приложению: обновления или пакеты обновлений могут помочь решить многие из проблем с совместимостью и дать возможность приложению работать в новой среде операционной системы.
* Апгрейд приложения до совместимого релиза: если более новая, совместимая версия приложения существует, лучшее решение — обновить до более новой версии.
* Изменение конфигурации безопасности: как пример, Защищенный режим Internet Explorer может быть смягчен, добавив сайт в список надежных сайтов или выключив Защищенный режим (что не рекомендуется).
* Запуск приложения в виртуализированной среде: если все другие методы недоступны, для решения проблем Вы можете запустить приложение в более раннем релизе Windows, используя инструменты виртуализации, такие как PC Microsoft Virtual и Microsoft Virtual Server.
* Использование функций совместимости приложения: проблемы приложения, такие как управление версиями операционной системы, могут быть смягчены, запуском приложения в режиме эмуляции. К этому режиму можно получить доступ, щелкнув правой кнопкой по ярлыку или .exe файлу и применяя режим эмуляции более ранней версии Windows на вкладки «**Совместимость**» (**Свойства -> Совместимость**). Так же, чтобы помочь в конфигурировании режима эмуляции с приложением, Вы можете использовать "**Мастер Совместимости Программ**". Эту функцию можно найти так: «**Панель управления» -> «Программы» -> «Выполнение программ, созданных для предыдущих версий Windows**».
* Выбор другого приложения, которое выполняет ту же самую функцию, но не имеет проблем с совместимостью: если другое совместимое приложение доступно, Вы можете использовать его.
  1. Проблемы перехода на новые версии программ. Мастер совместимости программ. Инструментарий учета аппаратных компонентов

Для многих организаций окончание срока действия поддержки ПО является настоящим кошмаром, связанным с увеличением уязвимости системы безопасности, потенциальными несоответствиями, влекущими за собой серьезные расходы, и ростом проблем, касающихся управления. Тем не менее, еще не поздно свести к минимуму такие риски и решить проблему перехода к новой версии Windows рационально и эффективно решать эти задачи постепенно в соответствии с бизнес-требованиями и приоритетами предприятия. Сформированная таким образом среда позволяет добиться гибкости в вопросах предоставления приложений и десктопов по запросу, максимально соответствует потребностям предприятий и пользователей и обеспечивает необходимую современным сотрудникам мобильность, предоставляя возможность работать в любое время, в любом месте и на любом устройстве.

Для перехода к более поздней версии ПО и трансформации предприятия ИТ-отделам и организации необходимо провести:

* Сокращение риска благодаря устранению опасных пробелов в системе безопасности и несоответствий, возникающих в результате продолжительного использования старого ПО;
* Увеличение продуктивности ИТ-отделов благодаря использованию виртуальных десктопов для перехода на последнее ПО;
* Экономия за счет доставки виртуальных приложений и десктопов на существующее аппаратное обеспечение
* Модернизация среды и предоставление сотрудникам возможности достигать большей эффективности при работе в любое время, в любом месте и на любом устройстве.

месте.

Режим совместимости программ Windows 10 позволяет запускать ПО на компьютере, которое нормально работало только в предыдущих версиях Windows, а в последней ОС программа не запускается или работает с ошибками. В этой инструкции о том, как включить режим совместимости с Windows 8, 7, Vista или XP в Windows 10 для исправления ошибок запуска программ.

По умолчанию, Windows 10 после сбоев в программах предлагает автоматически включить режим совместимости, однако только в некоторых из них и не всегда. Ручное же включение режима совместимости, которое раньше (в предыдущих ОС) выполнялось через свойства программы или ее ярлыка, теперь доступно не для всех ярлыков и иногда требуется использовать специальное средство для этого. Рассмотрим оба способа.

1. Использование динамически загружаемых библиотек
   1. Устранение проблем совместимости программного обеспечения.

Уменьшение проблем с совместимостьюприложения обычно зависит от различных факторов, таких как тип приложения и текущей поддержки приложения. Некоторые из общих методов включают следующее:

* Изменение конфигурации существующего приложения: Вы можете использовать инструменты, CompatibilityAdministrator или StandardUserAnalyzer, для обнаружения проблемы и создания исправления данного приложения, что решит проблему совместимости.
* Применение обновлений или пакетов обновлений к приложению: обновления или пакеты обновлений могут помочь решить многие из проблем с совместимостью и дать возможность приложению работать в новой среде операционной системы.
* Апгрейд приложения до совместимого релиза: если более новая, совместимая версия приложения существует, лучшее решение — обновить до более новой версии.
* Изменение конфигурации безопасности: как пример, Защищенный режим InternetExplorer может быть смягчен, добавив сайт в список надежных сайтов или выключив Защищенный режим (что не рекомендуется).
* Запуск приложения в виртуализированной среде: если все другие методы недоступны, для решения проблем Вы можете запустить приложение в более раннем релизе Windows, используя инструменты виртуализации, такие как PC Microsoft Virtual и Microsoft VirtualServer.
* Использование функций совместимости приложения: проблемы приложения, такие как управление версиями операционной системы, могут быть смягчены, запуском приложения в режиме эмуляции. К этому режиму можно получить доступ, щелкнув правой кнопкой по ярлыку или .exe файлу и применяя режим эмуляции более ранней версии Windows на вкладки «Совместимость».
* Выбор другого приложения, которое выполняет ту же самую функцию, но не имеет проблем с совместимостью: если другое совместимое приложение доступно, Вы можете использовать его.
  1. Конфигурирование программных и аппаратных средств.

Существуют различные классификации компьютерной техники:

* по этапам развития (по поколениям);
* по архитектуре;
* по производительности (микрокомпьютеры, в том числе персональные компьютеры; миникомпьютеры; мэйнфреймы (универсальныекомпьютеры);
* суперкомпьютеры.
* по условиям эксплуатации (офисные (универсальные) и специальные)
* по количеству процессоров;
* по потребительским свойствам и т.д.

Состав вычислительной системы называется ее конфигурацией. Различают аппаратную и программную конфигурацию. Современные компьютеры имеют блочную конструкцию. Аппаратную конфигурацию, необходимую для выполнения конкретных видов работ, можно собрать из готовых блоков и гибко изменять по мере необходимости.

Согласование между отдельными блоками выполняется с помощью устройств, называемых аппаратными интерфейсами. Стандарты на аппаратные интерфейсы называются протоколами. Аппаратные интерфейсы разделятся на последовательные и параллельные.

Последовательный интерфейс обеспечивает передачу данных последовательно, бит за битом и поэтому обеспечивают малую скорость передачи данных и имеют простое устройство.

Параллельные интерфейсы обеспечивают передачу данных одновременно группами битов, что повышает скорость передачи данных. Количество битов в группе называется разрядностью интерфейса. Существуют 8, 16, 32 и 64-разрядные интерфейсы.

В настоящее время базовая аппаратная конфигурация персонального компьютера включает следующие устройства:

* системный блок;
* монитор;
* клавиатуру;
* мышь.

Конечная цель выполнения любой программы – управление аппаратными средствами. Программное и аппаратное обеспечение работают в непрерывном взаимодействии, и их разделение является довольно условным. Между программами, также как между аппаратными средствами, существует взаимосвязь, поэтому можно говорить о программном интерфейсе. Программный интерфейс основан на протоколах – соглашениях о взаимодействии программ. Всё программное обеспечение вычислительной системы разбивается на несколько взаимодействующих между собой уровней (рис. 18). Каждый следующий уровень опирается на программное обеспечение предшествующих уровней. Такое разделение программного обеспечения упрощает разработку и эксплуатацию программ. Каждый следующий уровень повышает функциональные возможности всей системы.

Базовый уровень. Это самый низкий уровень программного обеспечения. Базовое программное обеспечение отвечает за взаимодействие с базовыми аппаратными средствами.

Системный уровень. Этот уровень обеспечивает взаимодействие прочих программ вычислительной системы с программами базового уровня и непосредственно с аппаратным обеспечением. От программ этого уровня во многом зависят эксплуатационные показатели всей вычислительной системы. При подключении к системе нового оборудования на системном уровне должна быть установлена программа, обеспечивающая взаимодействие других программ с этим оборудованием. Конкретные программы, отвечающие за взаимодействие с конкретными устройствами, называются драйверами устройств.

Служебный уровень. Программное обеспечение этого уровня взаимодействует как с программным обеспечением базового уровня, так и с программным обеспечением системного уровня. Служебные программы называются утилитами. Они предназначены для автоматизации работ по проверке, наладке и настройке вычислительной системы, а также для расширения и улучшения функций системных программ.

Прикладной уровень. Программное обеспечение этого уровня представляет собой комплекс прикладных программ, с помощью которых на данном рабочем месте выполняются конкретные работы. Диапазон возможных приложений вычислительной системы зависит от наличия прикладных программ для разных видов деятельности. Широта функциональных возможностей компьютера напрямую зависит от типа используемой операционной системы.

1. Решение проблем конфигурации с помощью групповых политик
   1. Групповые политики. Аутентификация. Учетные записи.

Групповая политика — это набор правил или настроек, в соответствии с которыми производится настройка рабочей среды приёма/передачи ([Windows](https://ru.wikipedia.org/wiki/Windows" \o "Windows), X-unix и другие операционные системы с поддержкой сети). Групповые политики создаются в [домене](https://ru.wikipedia.org/wiki/%D0%94%D0%BE%D0%BC%D0%B5%D0%BD_Windows_NT) и реплицируются в рамках домена. Объект групповой политики ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) Group Policy Object, GPO) состоит из двух физически раздельных составляющих: контейнера групповой политики ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) Group Policy Container, GPC) и шаблона групповой политики ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA" \o "Английский язык) Group Policy Template, GPT). Эти два компонента содержат в себе все данные о параметрах рабочей среды, которая включается в состав объекта групповой политики. Продуманное применение объектов GPO к объектам каталога [Active Directory](https://ru.wikipedia.org/wiki/Active_Directory) позволяет создавать эффективную и легко управляемую компьютерную рабочую среду на базе ОС [Windows](https://ru.wikipedia.org/wiki/Windows). Политики применяются сверху вниз по [иерархии каталога](https://ru.wikipedia.org/wiki/%D0%9A%D0%B0%D1%82%D0%B0%D0%BB%D0%BE%D0%B3_(%D1%84%D0%B0%D0%B9%D0%BB%D0%BE%D0%B2%D0%B0%D1%8F_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0)) Active Directory..

Групповые политики позволяют настраивать параметры для определенного набора пользователей или компьютеров внутри домена Active Directory. Также позволяют указать политики в одном месте для группы и применить к целевому набору пользователей.

Например, можно обеспечить применение стандартного набора настроек и конфигураций для групп пользователей или компьютеров в домене или по запросу. Во всех компаниях как правило есть различные отделы, например отдел системных администраторов, разработчиков, дизайнеров, каждому из отдела необходим свой стандартный набор программного обеспечения, их рабочие компьютеры должны быть сконфигурированы под специальные задачи и нужды. С помощью групповых политик можно создать наборы настроек для конкретных групп пользователей в домене. С помощью Active Directory GPO можно установить и управлять отдельными унифицированными наборами настроек, конкретно для дизайнеров или разработчиков.

Конфигурации для компьютеров или пользователей проще и эффективнее, т.к. расположены в одном месте и не требуют повтора на каждом компьютере.

Аутентификация – это процесс, в рамках которого подтверждается подлинность объекта. Под аутентификацией можно понимать процесс проверки подлинности введенного пользователем пароля, проверки подлинности электронного сообщения, проверки контрольной суммы файла. В рамках информационных систем перед процедурой аутентификации пользователь также проходит процедуру идентификации (определяется идентификатор, который однозначно идентифицирует пользователя) и авторизации (определяются права доступа, которые назначены пользователю). В данном случае аутентификация подтверждает, что идентификатор субъекта принадлежит конкретному пользователю. Наиболее популярными системами в интернете при работе, с которыми требуется прохождение процедуры аутентификации, являются электронная почта, интернет-банкинг, внутренние сайты компаний и интернет-магазины

Учетная запись или аккаунт на ПК означает совокупность личных сведений, которые пользователь вносит в компьютерную систему. Они доступны только ему. Это можно сравнить со своей комнатой, если вы проживаете в квартире с семьей. Как правило, учетка содержит данные, необходимые для идентификации личности в ходе подключения к системе, авторизации – логин и пароль.

В качестве логина обычно берется имя пользователя. Пароль же придумывают самостоятельно и состоит он из разных символов (цифр, букв, знаков). Для интернет-пользователей учетная запись то же самое, что Личный кабинет – онлайн-ресурс, где хранится персональная и прочая информация. Для большей защиты аккаунта от взлома часто предусмотрены дополнительные меры безопасности при прохождении аутентификации. К примеру, секретный вопрос, ответ на который знает только хозяин учетки.

Также в учетной записи часто содержатся дополнительные сведения о создателе странички: фотография, Ф. И. О., пол, псевдоним, национальность, группа крови, дата рождения, адрес проживания, № телефона и другие подробности.

1. Тестирование на совместимость в безопасном режиме
   1. Понятие совместимости программного обеспечения. Аппаратная и программная совместимость. Совместимость драйверов. Причины возникновения проблем совместимости. Методы выявления проблем совместимости программного обеспечения

Совместимость–способность аппаратных и программных средств работать с компьютерной системой. Совместимость– это способность компонентов взаимодействовать друг с другом.

Аппаратная (техническая) совместимость – способность одного компьютера работать с узлами, или устройствами, входящими в состав другого компьютера. Составной частью аппаратной совместимости является электромагнитная совместимость (ЭМС) – способность работающих (в том, числе автономно друг от друга) технических средств не создавать взаимных электромагнитных полей, а также функционировать при наличии внешних электромагнитных полей. ЭМС называют ограничение собственного электромагнитного излучения устройств до уровня, не влияющего на работу других устройств.

По аппаратной совместимости различают аппаратные платформы.

Аппаратная совместимость:

А) комплектующие, удовлетворяющие одному стандарту, являются взаимозаменяемыми;

В) различные части компьютера не конфликтуют между собой.

Программная совместимость – возможность выполнение одних и тех же программ на разных компьютерах с получением одинаковых результатов.

Совместимость программ – пригодность программ к взаимодействию друг с другом и в части к объединению в программные комплексы для решения более сложных задач.

Программная совместимость – это способность выполнения одинаковых программ с получением одних и тех же результатов. Программная совместимость:

А) программы, разработанные на одной машине, будут работать и на другой;

В) программы эмуляторы, позволяющие запускать ПО, разработанное для персональных компьютеров одного типа, на другом ПК.

Причины возникновения проблем совместимости:   
**-Несовершенство программного обеспечения**. Не все люди, которые занимаются написанием программного обеспечения, являются высококвалифицированными программистами. В основном это самоучки, начитавшиеся умных книг, знакомые с основами программирования. Написав программу, которая работает у него на компьютере, такой «программист» решает, что она будет работать и у других, совершенно не учитывая тот факт, что на компьютерах пользователей могут быть установлены разные операционные системы, компьютеры могут обладать разными ресурсами, устройствами, использовать разное программное обеспечение и т. д.

Это, конечно, совсем не означает, что профессионалы пишут программы без проблем: их преследуют те же ошибки. Однако квалифицированные программисты умеют адаптировать программу для работы в разных условиях, что делает ее менее уязвимой и, соответственно, более корректной в работе.

Очень часто бывает так, что человек садится за написание программы, не думая о том, что при работе она будет захватывать различные ресурсы, контролировать которые в одиночку не удастся. В результате появляется программное обеспечение, имеющее множество недочетов и постоянно выдающее ошибки.

Достаточно большой проблемой является и то, как программа распоряжается установленными в операционной системе системными библиотеками. Иногда системные библиотеки заменяются улучшенными версиями, с которыми отказываются работать другие программы, что, в свою очередь, также приводит к появлению ошибок.

**Несовершенство операционной системы**. Какими бы совершенными ни были операционные системы, они никогда не смогут создать нормальные условия для работы всего существующего программного обеспечения. Кроме того, совместимость операционных систем с выпуском каждой новой системы ухудшается. В таких условиях разработчики вынуждены писать программы, ориентированные на конкретную операционную систему.

Пользователю остается или работать с сертифицированными программами, или мириться со сложившейся ситуацией. Иногда и выбирать не приходится, ведь многие программы распространяются бесплатно (можно догадаться, какое у большой части из них качество работы).

**Отсутствие нужных ресурсов.** Часто даже небольшая программа требует для нормальной работы большее количество ресурсов, чем ей может предоставить операционная система. В таком случае для обеспечения работы программы система увеличивает файл подкачки, отбирающий оперативную память у других программ. Это приводит не только к снижению производительности компьютера, но и к появлению разнообразных ошибок.

Обычно такую проблему можно решить простым увеличением объема оперативной памяти. Однако это можно сделать и с помощью специальных утилит.

**-Ошибки в реестре.** Реестр — мозг операционной системы. Ошибки в нем негативно сказываются на всех процессах, происходящих в компьютере. Причиной возникновения сбоев в реестре являются все те же непрофессионально написанные программы, прописывающие свои файлы и ссылки в самых различных местах. Не стоит также забывать о троянских конях и «червях», которые только и мечтают о том, чтобы что-нибудь подпортить.

Для «лечения» реестра предназначены специальные утилиты, умеющие анализировать записи и удалять ошибочные и не используемые данные. При этом не следует забывать об элементарном сохранении рабочей версии файлов реестра.

**-Конфликты между устройствами**. Аппаратное обеспечение, как и программное, также не всегда бывает полностью совместимым. При использовании устройств доморощенных производителей вы обязательно получите целый букет подобных конфликтов. Например, перестанет работать звуковая карта или пропадет один из приводов компакт-дисков.

Прежде всего в такой ситуации следует проанализировать используемые конфликтующими устройствами прерывания и ресурсы. Для этого воспользуйтесь Диспетчером устройств или просмотрите отчеты об ошибках работы системы.

Достаточно часто в инструкции к устройству производители указывают несовместимые с ним комплектующие других производителей. Поэтому при сборке или модернизации компьютера обязательно внимательно изучите документацию к каждому из приобретенных устройств.

Часто подобные конфликты можно решить переустановкой платы расширения в другой слот или вручную, назначив им свободные ресурсы (с помощью Диспетчера устройств в свойствах конкретного устройства можно назначить прерывания и диапазон адресов). В случае с Windows Vista еще один действенный способ — замена проблемного устройства более новой моделью.

**-Вирусы, троянские кони и «черви».** Если вы пользуетесь Интернетом, то вам не избежать неприятных сюрпризов, таких как различные вирусы, трояны, «черви», программы-шпионы и т. д.

Общие методы уменьшения проблемы с совместимостью

Совместимы ли Ваши приложения с Windows 7, поможет определить подключение Application Compatibility Toolkit (ACT) 5.5. ACT также помогает определить, как будут влиять на Ваши приложения апгрейды. Так же Вы функции ACT могут использоваться для:

* Проверки своих приложений, устройств и компьютера на совместимость с новой версией операционной системы Windows
* Проверки совместимости обновления Windows
* Подключения в сообщество ACT и совместной оценки риска с другими пользователями ACT
* Тестирования своих Веб-приложений и Веб-сайтов на возможность проблем совместимости с новыми выпусками и обновлениями системы защиты Internet Explorer.

Методы уменьшения проблем с совместимостью

Уменьшение **проблем с совместимостью** приложения обычно зависит от различных факторов, таких как тип приложения и текущей поддержки приложения. Некоторые из общих методов включают следующее:

* Изменение конфигурации существующего приложения: Вы можете использовать инструменты, Compatibility Administrator или Standard User Analyzer (устанавливается с ACT), для обнаружения проблемы и создания исправления данного приложения, что решит проблему совместимости.
* Применение обновлений или пакетов обновлений к приложению: обновления или пакеты обновлений могут помочь решить многие из проблем с совместимостью и дать возможность приложению работать в новой среде операционной системы.
* Апгрейд приложения до совместимого релиза: если более новая, совместимая версия приложения существует, лучшее решение — обновить до более новой версии.
* Изменение конфигурации безопасности: как пример, Защищенный режим Internet Explorer может быть смягчен, добавив сайт в список надежных сайтов или выключив Защищенный режим (что не рекомендуется).
* Запуск приложения в виртуализированной среде: если все другие методы недоступны, для решения проблем Вы можете запустить приложение в более раннем релизе Windows, используя инструменты виртуализации, такие как PC Microsoft Virtual и Microsoft Virtual Server.
* Использование функций совместимости приложения: проблемы приложения, такие как управление версиями операционной системы, могут быть смягчены, запуском приложения в режиме эмуляции. К этому режиму можно получить доступ, щелкнув правой кнопкой по ярлыку или .exe файлу и применяя режим эмуляции более ранней версии Windows на вкладки «**Совместимость**» (**Свойства -> Совместимость**). Так же, чтобы помочь в конфигурировании режима эмуляции с приложением, Вы можете использовать "**Мастер Совместимости Программ**". Эту функцию можно найти так: «**Панель управления» -> «Программы» -> «Выполнение программ, созданных для предыдущих версий Windows**».

-Выбор другого приложения, которое выполняет ту же самую функцию, но не имеет проблем с совместимостью: если другое совместимое приложение доступно, Вы можете использовать его.

1. Cоздание в системе виртуальной машины для исполнения приложений
   1. Создание в системе виртуальной машины для исполнения приложений. Изменение настроек по умолчанию в образе. Подключение к сетевому ресурсу. Настройка обновлений программ. Обновление драйверов

В среде IT-специалистов виртуальной машиной принято называть Windows или Linux систему, которая функционирует на хосте ESX/ESXi. На самом деле это не совсем так. Виртуальная машина в действительности — это набор виртуального оборудования, собранного воедино для работы гостевой операционной системы. Аналогично тому, как физический компьютер все равно остается компьютером до установки операционной системы, так и виртуальная машина существует сама по себе до установки гостевой операционной системы. В обиходе виртуальными машинами довольно часто называют Windows или Linux системы. Если в этой книге встречается фраза “экземпляр гостевой операционной системы”, имейте в виду, что это экземпляры Windows, Linux или Solaris, установленные на виртуальную машину.

Для применения настроек к образу есть два способа.

Вы можете:

Настроить только системные параметры, а пользовательские импортировать с помощью REG-файлов в настроенный WIM-образ (либо в исходный образ еще до начала описываемой здесь установки).

Настроить как пользовательские, так и системные параметры, а затем обеспечить с помощью файла ответов копирование профиля встроенной учетной записи Администратор в профиль стандартного пользователя.

Подключение к сетевому ресурсу:

Для этого можно воспользоваться командой, которую следует ввести в командной строке: net use <буква диска>:

Например: net use f: [\\dimapc\dima](file:///\\dimapc\dima)

Кроме того, подключение к общему ресурсу реализуется средствами программы Проводник (Explorer).

* Запустите программу Проводник (Explorer) (Пуск > Программы > Проводник (Start > Programs > Windows Explorer)).
* В меню Сервис (Tools) выберите параметр Подключить сетевой диск (Map Network Drive).
* Выберите букву сетевого диска и введите путь к общему ресурсу или выберите ресурс из списка доступных в сети общих ресурсов.
* Щелкните на кнопке OK.

Преимуществом использования команды net use является возможность подключения к скрытым общим ресурсам, то есть к ресурсам, имена которых завершаются символом “$” (к таким ресурсам можно подключится и с помощью программы Проводник, но с указанием точного сетевого адреса ресурса). Кроме того, команда net use применяется и в командных файлах.

Первым делом необходимо включить автоматическое обновление. Для этого нужно войти в «Панель управления» и найти в нем пункт «Центр обновления Windows». Включение автоматического обновления Windows В следующем окне необходима настройка. Выбираем «Настройка параметров». Настройка параметров обновления. Рекомендуем продублировать настройки, в соответсвии с тем, что отображены на изображении ниже. Детальная настройка параметров обновления Windows. На пиктограмме со щитом с галочкой имеется раскрывающийся список. Выбор типа обновления Windows. Важно, чтобы был выбран один из первых трех пунктов. Лучшим считается автоматический процесс, который и рекомендуется системой. При выборе последнего пункта на безопасность ОС рассчитывать на 100% нельзя. Настроив параметры обновления, нажимается кнопка «OK».

Обновление драйвера устройства

* Введите "диспетчер устройств" в поле поиска на панели задач и щелкните элемент Диспетчер устройств.
* Выберите категорию, чтобы просмотреть имена устройств, а затем щелкните правой кнопкой мыши (или нажмите и удерживайте) устройство, которое требуется обновить.
* Выберите пункт Автоматический поиск обновленных драйверов.
* Выберите обновить драйвер.
* Если Windows не найдет новый драйвер, можно попытаться его найти на веб-сайте изготовителя устройства и выполнить соответствующие инструкции.

1. Создание образа системы
   1. Создание образа системы. Восстановление системы. Разработка модулей программного средства. Настройка сетевого доступа.

Образы системы содержат все данные, хранящиеся на вашем компьютере, по состоянию на определенный момент времени.

* Щелкните правой кнопкой мыши кнопку
* Начните, а затем выберите Панель управления > система и обслуживание >резервное копирование и восстановление.
* В левой области выберите. Создать системное изображениеи следуйте шагам мастера. Если вам будет предложено ввести пароль администратора или подтверждение, введите пароль или подведите подтверждение.

Примечание:Чтобы создать образ системы на основе диска, этот диск должен быть отформатирован для файловой системы NTFS. Если вы хотите сохранить образ системы на жестком диске или на USB-устройстве флэш-памяти, то они должны быть отформатированы для файловой системы NTFS.

"Восстановление системы" — это средство ® Windows® предназначенное для защиты и восстановления программного обеспечения компьютера. Восстановление системы делает "моментальный снимок" некоторых системных файлов и реестра Windows и сохраняет их в качестве точек восстановления. При сбое установки или повреждения данных восстановление системы может вернуть систему в рабочее состояние без необходимости переустановки операционной системы. Оно восстанавливает среду Windows, возвращаясь к файлам и настройкам, сохраненным в точке восстановления. Программа по умолчанию создает точки восстановления один раз в день. Она постоянно отслеживает действия в системе и создает точку восстановления при определенных действиях. К типам действий, которые запускают автоматическое создание точек восстановления, относятся: установка программного обеспечения

* обновление драйверов оборудования
* установка новых драйверов оборудования
* ручное создание точек восстановления

Практически все сложное программное обеспечение имеет модульную структуру, то есть состоит из программных модулей, при этом [разработка программных модулей программного обеспечения](https://www.artwell.ru/services/individualnyy-sayt-portal/) может быть выполнена:

* Как составляющая часть процесса разработки ПО. В данном случае разработка модулей, в зависимости от проекта, может вестись последовательно и (или) параллельно, но всегда завершается их объединением в единую систему.
* В процессе доработки, усовершенствования или [модернизации ранее созданного ПО](https://www.artwell.ru/services/software_modules_development_and_integration/). Модульная конструкция программного обеспечения позволяет вносить изменения в отдельные модули, удалять некоторые из них или дополнять систему новыми модулями. При этом разработка программных модулей программного обеспечения и их внедрение, как и изменение или удаление, не заставляет переделывать всю систему ПО в целом.

Для открытия общего доступа к диску (папке) в Windows 10, выполните следующие действия:

* кликните правой кнопкой мыши на иконке диска (папки) и в выпадающем меню выберите «Свойства»;
* затем перейдите на вкладку «Доступ» и нажмите на кнопку «Расширенная настройка…»;
* в открывшемся окне, установите галочку напротив надписи «Открыть доступ к этой папке»
* в поле «Имя общего ресурса» введите сетевое имя (необходимо для удобства идентификации ресурса в домашней локальной сети)
* после чего кликнете «Применить».

1. Настройка обновлений программ
   1. Установка и настройка антивируса. Настройка обновлений с помощью зеркала

Одной из самых важных программ в компьютере является антивирус.

От правильного выбора, установки и настройки антивируса зависит безопасность наших данных. Это, наверное, одна из немногих программ, к выбору которой следует отнестись очень серьезно. Все мы знаем, что если компьютер не защищен хорошим антивирусом и файрволом, в Интернет лучше не выходить. Выбор антивирусов достаточно велик: есть платные антивирусы, есть бесплатные, есть также и лжеантивирусы.

Идеальный вариант – сразу после установка операционной системы устанавливаете антивирус, а затем все остальные программы. Такая установка даст возможность антивирусу осуществлять контроль за всеми программами, установленными на компьютере, и обеспечит максимальную защиту. Переустанавливать антивирус нежелательно. Также нежелательно устанавливать антивирус в последнюю очередь. Во-первых, это значительно понижает защиту компьютера и контроль над программами. Во-вторых, вы рискуете поймать вирус, а если компьютер уже заражен, то и антивирус тоже будет заражен при установке.

**Настройка:**

* Настройки «по умолчанию» оставлять не стоит, они не идеальны и часто именно из-за этого можно поймать вирус. Лучше всего делать индивидуальную настройку на максимальную защиту и ежедневное обновление антивирусных баз. Компьютер при этом будет работать немного медленнее, но тут каждый сам решает, что важнее – защита или быстродействие.
* Полную проверку можно делать раз в неделю-две, а быструю проверку нужно делать ежедневно. Быстрая проверка – это проверка отдельных файлов и папок. «По умолчанию» проверяются: системная память, объекты автозапуска и загрузочные сектора. К этому списку нужно добавить еще две директории, в которые чаще всего проникают вирусы: C:\Documents and Settings и C:\WINDOWS\system32.

Зеркало обновлений – это компьютер, настроенный как источник обновлений для антивируса на других ПК в локальной сети. Используется в сетях, где не все ПК имеют доступ в Интернет.

Для создания зеркала обновлений:

* Откройте меню антивируса, щёлкнув по значку в трее и выбрав Центр безопасности;
* Щёлкните по значку замка, разрешите внесение изменений и нажмите кнопку с шестерёнкой (Настройки)
* Перейдите на вкладку Обновление и выберите Дополнительные настройки.
* Установите переключатель Зеркало обновлений в положение Вкл. Открывается окно настройки зеркала:
* Нажмите Обзор и укажите папку, куда будут сохраняться обновления. Она будет источником обновления для остальных ПК в локальной сети.
* Адрес – адрес данного ПК в локальной сети, который будет использоваться как адрес для обновления через антивирусную сеть другими ПК в этой сети. Порт – номер порта, который будет использоваться для раздачи обновлений.

1. Оптимизация использования памяти и жесткого диска
   1. Оптимизация использования памяти. Оптимизация использования жесткого диска. Оптимизация использования сети. Инструменты повышения производительности программного обеспечения.

Оптимизация использования памяти может преследовать две цели – увеличение объема памяти, доступной приложениям, и повышение быстродействия обращений к памяти. На старых машинах, когда объем установленной физической памяти не превышал единиц мегабайт, зачастую приходилось чем-то жертвовать; в современных системах достижение обеих целей уже не противоречит друг другу. Нет сомнений в том, что, чем больше установленный объем ОЗУ, тем лучше – компьютер будет мощнее: он позволит загружать приложения, особо «жадные» до памяти; увеличить число одновременно работающих приложений (в многозадачной ОС); ускорить работу. Однако для обеспечения возможности использования памяти в ряде случаев требуются некоторые дополнительные действия по конфигурированию компьютера.  
В системах с размером установленной памяти более 640 Кбайт возможны различные варианты использования последних 384 Кбайт из первого мегабайта физической памяти:

* память не используется;
* область (или часть ее) перемещается в конец дополнительной памяти;
* область (или часть ее) используется в качестве теневой памяти адаптеров и ROM BIOS.

Наверно все пользователи хотят, чтобы их компьютеры могли выполнять требуемые действия намного быстрее, чем в текущий момент. Процесс обработки информации можно ускорить разными способами. К подобным вариантам относится оптимизация работы винчестера. Для подобного процесса в Виндовс имеется огромное количество настроек и инструментов. Давайте рассмотрим, как ускорить работу жесткого диска и в целом работу компьютера.

* Отключение индексирования
* Отказываемся от ведения журнала
* Очистка от ненужных файлов
* Дефрагментация
* Переразметка
* Перемещение файлов на съемный носитель
* Отключение автозагрузки программ
* Обновление драйвера контроллера диска

Современные ЭВМ обладают очень большой мощностью. Скорость работы процессора (ЦП) современных ЭВМ измеряется гигагерцами, объём оперативной памяти гигабайтами, а современные интерфейсы устройств обеспечивают скорость обмена данными порядка, как минимум, нескольких сотен мегабайт в секунду. Производительность, которая ещё несколько лет назад казалась «сказочной» в настоящее время стала нормой жизни.

Однако параллельно росту мощности ЭВМ увеличивается и ресурсоёмкость приложений. У приложений совершенствуется функционал, интерфейс, возрастает объём обрабатываемых данных и как следствие системные требования. Поэтому вопрос об увеличении быстродействия приложений не теряет своей актуальности.

Общие вопросы быстродействия программ.

Быстродействие программ (ПО) зависит от многих факторов, но основными из них являются два:

* Соотношение между реальными системными требованиями ПО и существующей аппаратной конфигурацией ЭВМ;
* Алгоритмы работы ПО.

Если низкое быстродействие обусловлено первым фактором, то решением является модернизация аппаратной части (hardware). В некоторых случаях проблему можно решить также с помощью тонкой настройки hardware и операционной системы. Однако этот путь имеет ряд недостатков:

* Увеличивается производительность hardware, а вовсе не быстродействие ПО;
* Производительность hardware ограничена возможностями существующих в данный момент элементной базы и инженерных решений в данной области;
* Большие финансовые затраты на модернизацию и настройку по причине высокой стоимости комплектующих ЭВМ и услуг специалистов требуемой квалификации.

По этим причинам при разработке ПО прибегают к увеличению его быстродействия с помощью различных средств программной инженерии. Это позволяет:

* Обеспечить работу нового ПО на уже существующем оборудовании;
* Разработать масштабируемое ПО;
* Значительно уменьшить финансовые и трудовые затраты при внедрении.

Вместе с тем и у этого пути имеется ряд недостатков:

* Значительно усложняется процесс разработки ПО, так как более «быстрые» алгоритмы сложнее более «медленных» (на пример алгоритм бинарного поиска сложнее, чем алгоритм линейного поиска);
* Реализация более сложных алгоритмов, как правило, требует привлечения специалистов более высокой квалификации;
* В случае работы с большими объёмами данных или выполнении задач требующих больших и сложных вычислений, ресурсоёмкость ПО всё равно остаются достаточно высокой. Несмотря, на какие либо способы увеличения быстродействия.

Таким образом, в общем случае обеспечение быстродействия ПО является комплексной задачей.

Однако следует заметить, что среди существующих задач, очень немногие обладают высокой ресурсоёмкостью. Вследствие этого в большинстве случаев не требуется никаких действий относительно hardware и требуемого результата можно достичь, прибегая только к программной инженерии.

Программная инженерия предоставляет несколько способов увеличения быстродействия программ. Рассмотрим их на примере языков программирования Delphi и Assembler.

Увеличение быстродействия программ.

Как было показано в предыдущем параграфе, можно увеличить быстродействие ПО соответствующим образом реализовав его алгоритмы. Количественным показателем быстродействия алгоритма (а, следовательно, и ПО) является время его выполнения, измеренное по специальной методике, так называемого профилирования. Таким образом, в общем случае выбор наиболее «быстрых» алгоритмов сводится к измерению времени их выполнения и сравнении полученных результатов между собой. Такой способ анализа быстродействия является наиболее объективным. На протяжении многих лет программистами был накоплен большой опыт профилирования, который позволяет сделать определённые выводы относительно возможности оптимизации быстродействия ПО ещё на стадии написания.

Эти выводы были обобщены и представлены в виде определённых рекомендаций. Если программист будет следовать данным рекомендациям, то написанная программа вероятнее всего будет обладать большим быстродействием, чем в случае их игнорирования. Однако следует ещё раз подчеркнуть, что достоверные сведения о быстродействии может дать только профилирование. Это обусловлено тем, что быстродействие алгоритма определяет в первую очередь его конкретная реализация. Кроме того необходимо ещё раз отметить, что в отношении увеличения быстродействия ПО программная инженерия не всесильна.

В чём же состоят выше упомянутые рекомендации? Их краткое содержание применительно к языку программирования Delphi приведено ниже.

* При написании кода программ рекомендуется избегать процедур, состоящих из сотен строк. Практически всегда в них можно выделить блоки, которые лучше оформить в виде отдельной процедуры. Возможно, позже вы ей даже воспользуетесь где-то в другом месте. Не говоря уже о том, что это повышает понимание программы и вами, и другими программистами. К тому же так проще искать «узкие» места в программе.
* Использование оператора case (switch) вместо многократных if… then… else (if… else). Во втором варианте компилятор будет выполнять проверку условия столько раз, сколько у вас вариантов. В первом проверка выполняется лишь однажды.
* Некоторые действия могут быть довольно продолжительными, поэтому рекомендуется выносить за рамки цикла всё, что можно выполнить вне его, чтобы избежать большого числа повторений внутри цикла.
* В циклах типа for нужно стараться, чтобы значение счетчика уменьшалось до нуля, а не наоборот — начиналось с нуля. Это связано с особенностями процессора. Сравнение с нулём выполняется гораздо быстрее, чем с другим числом.
* Пользоваться типом Variant только при необходимости. Операции над этим типом сложнее, чем, например, над Integer или String.
* Не злоупотреблять «программированием на компонентах». В частности не использовать компонент TTreeView для хранения древовидных структур данных — он работает очень медленно и предназначен только для визуального отображения. В случае работы со структурами данных лучше использовать алгоритмы, созданные самостоятельно на основе фундаментальных.
* Сохранение и загрузка свойств компонентов с помощью методов ReadComponent и WriteComponent работает довольно медленно, поэтому по возможности рекомендуется сохранять и восстанавливать состояние программы между сеансами при помощи других способов.
* Заменить простой в реализации алгоритм на более сложный, но с большим быстродействием. Например, если заранее известно, что в списке для поиска будет много элементов, лучше его отсортировать и применять бинарный поиск вместо линейного.
* В критических с точки зрения быстродействия местах программы делать вставки на ассемблере. Команды ассемблера напрямую транслируются в машинный код. Таким образом, в отличие от высокоуровневых языков при компиляции отсутствует проблема синхронизации и ряд других негативных обстоятельств.

Для других языков программирования вышеприведённый список может несколько отличаться, в частности отсутствием поддержки ассемблера и как следствие возможности оптимизации с его помощью (Java, Visual C# и др.).

Особо следует отметить, что рекомендации 3 и 4 применяются не только для языков высокого уровня, но и для ассемблера. Помимо вышеуказанных для увеличения быстродействия программ написанных на ассемблере, в том числе и вставок, существуют следующие рекомендации:

* Замещение универсальных инструкций на учитывающие конкретную ситуацию, например, замена умножения на степень двойки на команды сдвига (отказ от универсальности).
* Уменьшение количества передач управления в программе: за счет преобразования подпрограмм в макрокоманды для непосредственного включения в машинный код; за счет преобразования условных переходов так, чтобы условие перехода оказывалось истинным значительно реже, чем условие для его отсутствия; перемещение условий общего характера к началу разветвленной последовательности переходов; преобразование вызовов, сразу за которыми следует возврат в программу, в переходы («сращивание хвостов» и «устранение рекурсивных хвостов») и т.д.
* Максимальное использование всех доступных регистров за счет хранения в них рабочих значений всякий раз, когда это возможно, чтобы минимизировать число обращений к памяти, упаковка множественных значений или флагов в регистры и устранение излишних продвижений стека (особенно на входах и выходах подпрограмм).

ЗАКЛЮЧЕНИЕ

Производственная практика является одной из неотъемлемых частей подготовки квалифицированных специалистов всех направлений. Во время прохождения практики происходит закрепление и конкретизация результатов теоретического обучения, приобретение студентами умения и навыков практической работы по избранной специальности и присваиваемой квалификации.

В работе были рассмотрены технологии разработки программных систем на основе CASE технологий. Подробно разобрано определение такого широкого понятия как CASE средство, определены главные составляющие CASE продукта.

Развитие информационных технологий сегодня диктует новый уровень сложности востребованных информационных систем. Крупные проекты сегодня требуют современных решений, характеризуются факторами, требующими современных методов моделирования. Такого рода разработка программных систем не возможна в полной мере своей эффективности без использования CASE средств. Современные CASE-инструменты охватывают обширную область поддержки многочисленных технологий проектирования ИС: от простых средств анализа и документирования до полномасштабных средств автоматизации, покрывающих весь жизненный цикл ПО.

В процессе прохождения производственная практики были проделаны следующие работы и получен следующий опыт:

* интеграция модулей в программное обеспечение;
* отладка программных модулей.

умения:

* использования выбранной системы контроля версий;
* использования метода для получения кода с заданной функциональностью и степенью качества
* модели процесса разработки программного обеспечения;
* основные принципы процесса разработки программного обеспечения;
* основные подходы к интегрированию программных модулей;
* основы верификации и аттестации программного обеспечения.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ И ЛИТЕРАТУРЫ

**Основные источники:**

1. Фёдорова, Г. Н. Сопровождение информационных систем: учебник для студ. учреждений сред.проф. образования / Г. Н. Фёдорова. – Москва: Издательский центр «Академия», 2018. – 320 с. ISBN 978 -5-4468-7029-5

**Дополнительные источники:**

1. Федорова, Г. Н. Разработка, внедрение и адаптация программного обеспечения отраслевой направленности: учеб.пособие / Г.Н. Федорова. – М.: КУРС: ИНФРА-М, 2017. – 336 с. (Среднее Профессиональное Образование);
2. Федорова, Г. Н. Разработка, внедрение и адаптация программного обеспечения отраслевой направленности: Учебное пособие. / Федорова Г.Н. – М.: КУРС, НИЦ ИНФРА-М, 2016. – 336 с.: 60x90 1/16. – (Среднее профессиональное образование) (Переплёт 7БЦ) ISBN 978-5-906818-41-6;
3. Исаченко, О. В. Программное обеспечение компьютерных сетей: Учебное пособие / О.В. Исаченко. – М.: ИНФРА-М, 2016. – 117 с.: 60x88 1/16. – (Среднее профессиональное образование). (обложка) ISBN 978-5-16-004858-1;
4. Шаньгин, В. Ф. Информационная безопасность компьютерных систем и сетей: учеб.пособие / В.Ф. Шаньгин. – М.: ИД «ФОРУМ» : ИНФРА-М, 2017. –416 с. – (Профессиональное образование).

**Интернет-ресурсы:**

1. Учебники по программированию <http://programm.ws/index.php>;
2. От модели объектов - к модели классов. Единое окно доступа к образовательным ресурсам. <http://real.tepkom.ru/Real_OM-CM_A.asp>.