Министерство образования Калининградской области

государственное бюджетное учреждение Калининградской области

профессиональная образовательная организация

«Колледж информационных технологий и строительства»

(ГБУ КО ПОО «КИТиС»)

**Отчет по учебной практике**

УП.04 Сопровождение и обслуживание программного обеспечения компьютерных систем

по ПМ.04 Сопровождение и обслуживание программного обеспечения компьютерных систем

Специальность 09.02.07 «Информационные системы и программирование»

Сроки прохождения практики:

с «13» февраля 2023 г. по «4» марта 2023 г.

Место практики ГБУ КО ПОО «КИТиС»

|  |  |
| --- | --- |
| Выполнил: | студент 2 курса,  группы ИСп 21-2К  Соколов Ростислав Александрович  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) |
| Проверила: | Большакова-Стрекалова Анна Викторовна  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (оценка)  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись, дата) |

Калининград, 2023

Введение

1 Обеспечение внедрения и поддержки программного обеспечения компьютерных систем

* 1. Инструктаж по технике безопасности
  2. Описание выбранного аппаратного и программного обеспечения
  3. Методы внедрения и анализа функционирования программного обеспечения
  4. Сценарий и техническое задание на внедрение ПО
  5. Загрузка и установка программного обеспечения
  6. Настройка отдельных компонентов программного обеспечения компьютерных систем и конфигурирование

1. Проблемы совместимости и методы их устранения
   1. Тестирование на совместимость в безопасном режиме
   2. Инструменты повышения производительности программного обеспечения
   3. Средства диагностики оборудования
   4. Измерение и анализ эксплуатационных характеристик качества программного обеспечения
   5. Оптимизация и модификация ПО
   6. Разработка руководства оператора
2. Этапы сопровождения и обслуживания ПО
   1. Разработка технического задания на сопровождение
   2. Обеспечение качества компьютерных систем в процессе эксплуатации
   3. Разработка сопровождающей документации
3. Методы и средства защиты компьютерных систем
   1. Тестирование на ошибки ввода/вывода данных
   2. Тестирование интерфейса
   3. Тестирование контента
   4. Анализ рисков. Выявление первичных и вторичных ошибок
   5. Обнаружение вируса и устранение последствий его влияния
   6. Тестирование защиты программного обеспечения
   7. Работа с реестром. Работа с программой восстановления файлов и очистки дисков
4. Индивидуальное задание
   1. Практические работы
   2. Методы верификации программного обеспечения
   3. Анализ работы инсталляторов, мастеров установки, архиваторов

Заключение

Список использованных источников

Введение

Учебная практика является составной частью учебно-воспитательного процесса и имеет важное значение в подготовке квалифицированного специалиста.

Целью учебной практики является закрепление, расширение и углубление полученных теоретических знаний, приобретение практических навыков самостоятельной работы, выработку умений применять их при решении конкретных социально-экономических вопросов и принятии управленческих решений, способствует комплексному формированию общекультурных, общепрофессиональных и профессиональных компетеций обучающихся.

Задачами учебной практики являются:

- закрепление приобретенных теоретических знаний;

- закрепление практических навыков самостоятельной работы, навыков самостоятельного формулирования выводов, полученных по результатам работы;

- приобретение практических навыков в будущей профессиональной деятельности или в отдельных ее разделах;

- закрепление практических приемов, методов, и способов обработки, представления и интерпретации результатов проведенных практических исследований;

- закрепление навыков презентации и защиты результатов выполняемой работы.

1 Обеспечение внедрения и поддержки программного обеспечения компьютерных систем

* 1. Инструктаж по технике безопасности

1. Перед работой:
   1. Необходимо подготовить рабочее место.
   2. Отрегулировать освещение и убедиться в отсутствии бликов на экране.
   3. Проверить правильность подключения оборудования к электросети.
   4. Проверить исправность проводов питания и отсутствие оголенных участков проводов.
   5. Убедиться в наличии заземления системного блока, монитора и защитного экрана.
   6. Протереть антистатической салфеткой поверхность экрана монитора и защитного экрана.
2. Во время работы:
   1. Работнику при работе на ПК запрещается:

- прикасаться к задней панели системного блока (процессора) при включенном питании;

- переключать разъемы интерфейсных кабелей периферийных устройств при включенном питании;

- допускать попадание влаги на поверхность системного блока (процессора), монитора, рабочую поверхность клавиатуры, дисководов, принтеров и других устройств;

- производить самостоятельное вскрытие и ремонт оборудования;

- работать на компьютере при снятых кожухах;

- отключать оборудование от электросети и выдергивать электровилку, держась за шнур.

* 1. Продолжительность непрерывной работы с компьютером без регламентированного перерыва не должна превышать 2-х часов.
  2. Во время регламентированных перерывов с целью снижения нервно - эмоционального напряжения, утомления зрительного анализатора, устранения влияния гиподинамии и гипокинезии, предотвращения развития познотонического утомления выполнять комплексы упражнений.

1. Во время аварийных ситуациях:
   1. Во всех случаях обрыва проводов питания, неисправности заземления и других повреждений, появления гари, немедленно отключить питание и сообщить об аварийной ситуации руководителю.
   2. Не приступать к работе до устранения неисправностей.
   3. При получении травм или внезапном заболевании немедленно известить своего руководителя, организовать первую доврачебную помощь или вызвать скорую медицинскую помощь.
2. По окончанию работы:
   1. Отключить питание компьютера.
   2. Привести в порядок рабочее место.
   3. Выполнить упражнения для глаз и пальцев рук на расслабление.

1.2 Описание аппаратного обеспечения.

Blender 

Blender — профессиональное свободное и открытое программное обеспечение для создания трёхмерной компьютерной графики, включающее в себя средства моделирования, скульптинга, анимации, симуляции, рендеринга, постобработки и монтажа видео со звуком, компоновки с помощью «узлов», а также создания 2D-анимаций. В настоящее время пользуется большой популярностью среди бесплатных 3D-редакторов в связи с его быстрым стабильным развитием и технической поддержкой.

Возможности данного ПО:

* Поддержка разнообразных геометрических примитивов.
* Универсальные встроенные механизмы рендеринга и интеграция с внешними рендерерами YaRay, LuxRender и многими другими.
* Динамика мягких/твердых тел на основе физических тел Bullet.
* Система частиц включающая в себя систему волос на основе частиц.
* Модификаторы для применения неразрушающих эффектов.
* Язык программирования Python используется как средство определения интерфейса, создания инструментов и прототипов, системы логики в играх, как средство импорта/экспорта файлов, автоматизации задач.
* Базовые функции нелинейного видео и аудио монтажа.
* Композитинг видео, работа с хромакеем.
* [Трекинг](https://ru.wikipedia.org/wiki/%D0%A2%D1%80%D0%B5%D0%BA%D0%B8%D0%BD%D0%B3_(%D0%B2%D0%B8%D1%80%D1%82%D1%83%D0%B0%D0%BB%D1%8C%D0%BD%D0%B0%D1%8F_%D1%80%D0%B5%D0%B0%D0%BB%D1%8C%D0%BD%D0%BE%D1%81%D1%82%D1%8C)) камеры и объектов.
* Real-time контроль во время физической симуляции и рендеринга.
* Процедурное и node-based текстурирование, а также возможность рисовать текстуру прямо на модели.
* Grease Pencil — инструмент для 2D-анимации в полном 3D-пайплайне.

Достоинства Blender:

* Программа распространяется бесплатно.
* Способность к решению многих задач 3Д моделирования.
* Необычный, но удобный способ размещения объектов.
* Возможность анимирования персонажа.
* Способность создания эффекта течения воды.
* Гибкий инструментарий создания анимаций.
* Возможность быстрого и качественного создания реалистичных визуализаций.

Требования:

|  |  |  |
| --- | --- | --- |
|  | Минимальные | Рекомендуемые |
| Процессор | Intel core i3 | Intel core i9 |
| Объем RAM | 2 ГБ | 32 ГБ |
| Видеокарта | 2GB RAM, OpenGL 4.3 | 8GB RAM |
| ОС | Linux, MacOS, Windows, SteamOS | |

1.3 Методы внедрения и анализа функционирования программного обеспечения.

Основные этапы внедрения программного продукта:

* 1. Обследование – диагностика предприятия или его обследование. Под обследованием подразумевается диагностика всех процессов. Которые будет охватывать будущая система.
  2. Разработка ТЗ – включает в себя описание всех справочников системы, всех алгоритмов расчета, отчетных форм, пользователей и описание разграничения прав доступа пользователей.
  3. Настройка системы – формирование в программе всех справочников системы, настройка всех алгоритмов расчета, форм ввода и отчетных форм, ввод пользователей системы и настройка прав доступа.
  4. Тестирование системы – подготовка демонстрационного примера, внесение тестовых данных, проверку алгоритмов расчета и исправление обнаруженных проблем.
  5. Опытная эксплуатация – работа с реальными данными, но при этом параллельно используется прежняя старая система. Этот этап необходим длят ого, чтобы можно было сопоставить результаты работы в новой системе с результатами, которые получены были прежним способом
  6. Промышленная эксплуатация – переход на новый программный продукт и отказ от всех альтернативных способов работы за рамками данной системы.

1.4 Сценарий и техническое задание на внедрение ПО.

Сценарии внедрения - это сводка задач внедрения продукта.

Внедрение программного обеспечения — процесс настройки программного обеспечения под определённые условия использования, а также обучения пользователей работе с программным продуктом.

1.5 Загрузка и установка программного обеспечения.

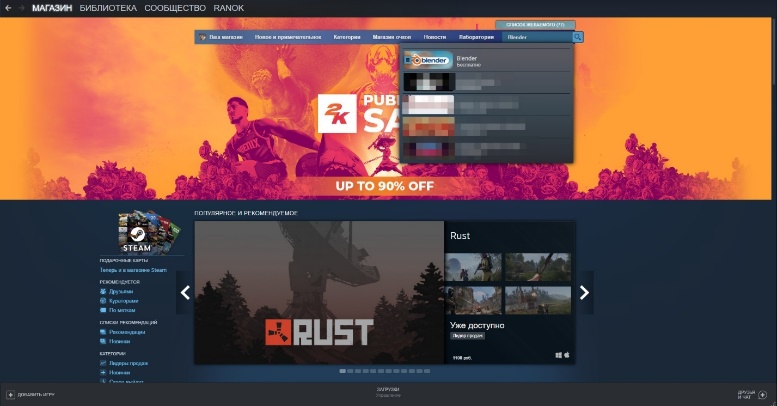
Загрузить и установить Blender можно в специальном приложении Steam. Заходим в меню магазин и в поиске пишем Blender:

Рисунок 1 – Поиск ПО в меню магазина

Выбираем нужное для нас приложение и нажимаем кнопку «Бесплатно»:

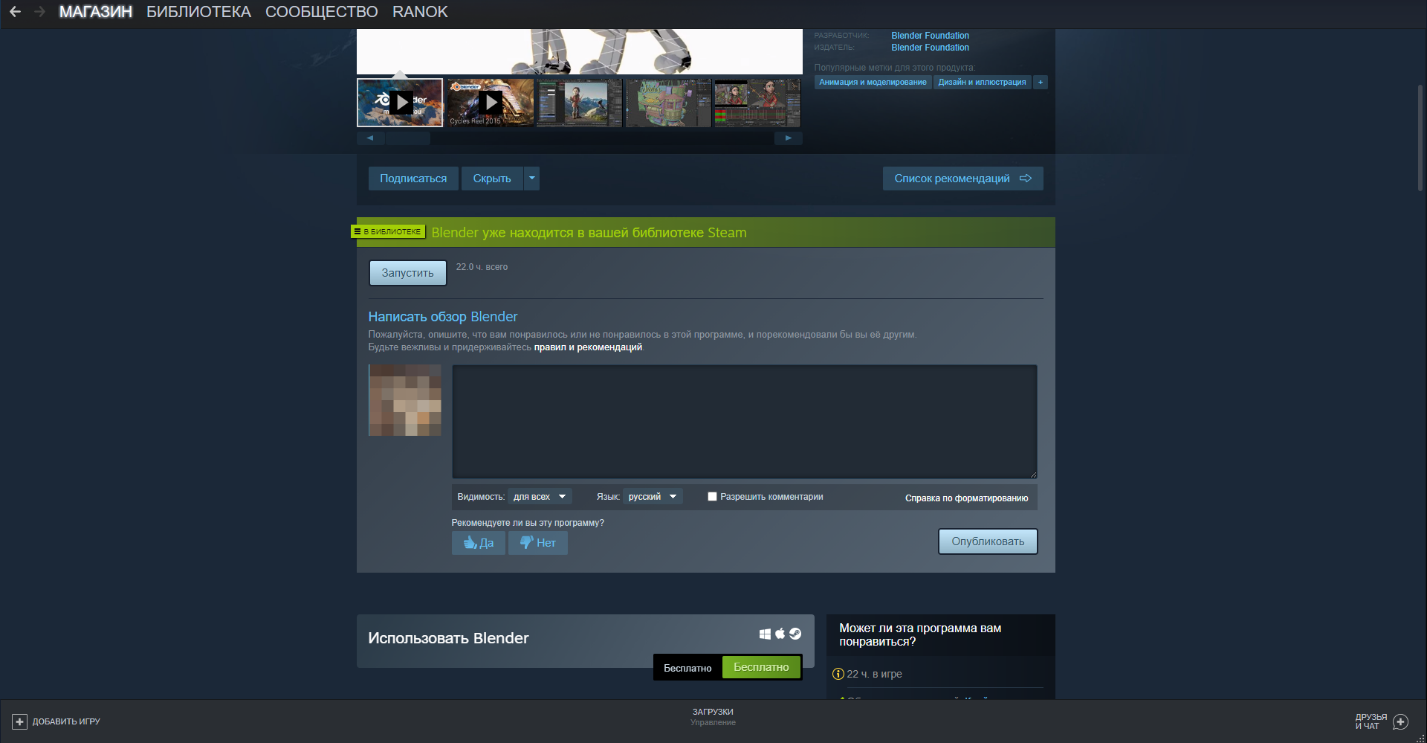
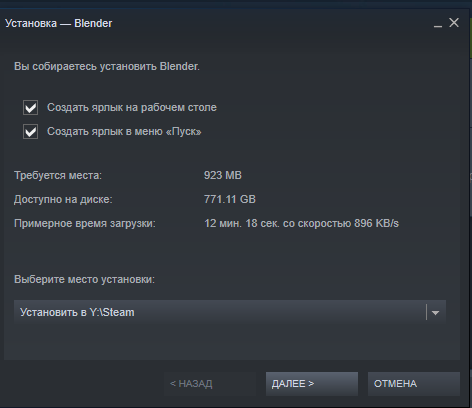
Рисунок 2   
  
Выбираем путь установки:

Рисунок 3 - Выбор пути файла и установка.

Ждем, когда загрузка пройдет нажимаем кнопку «Готово» и ждем, когда загрузится:

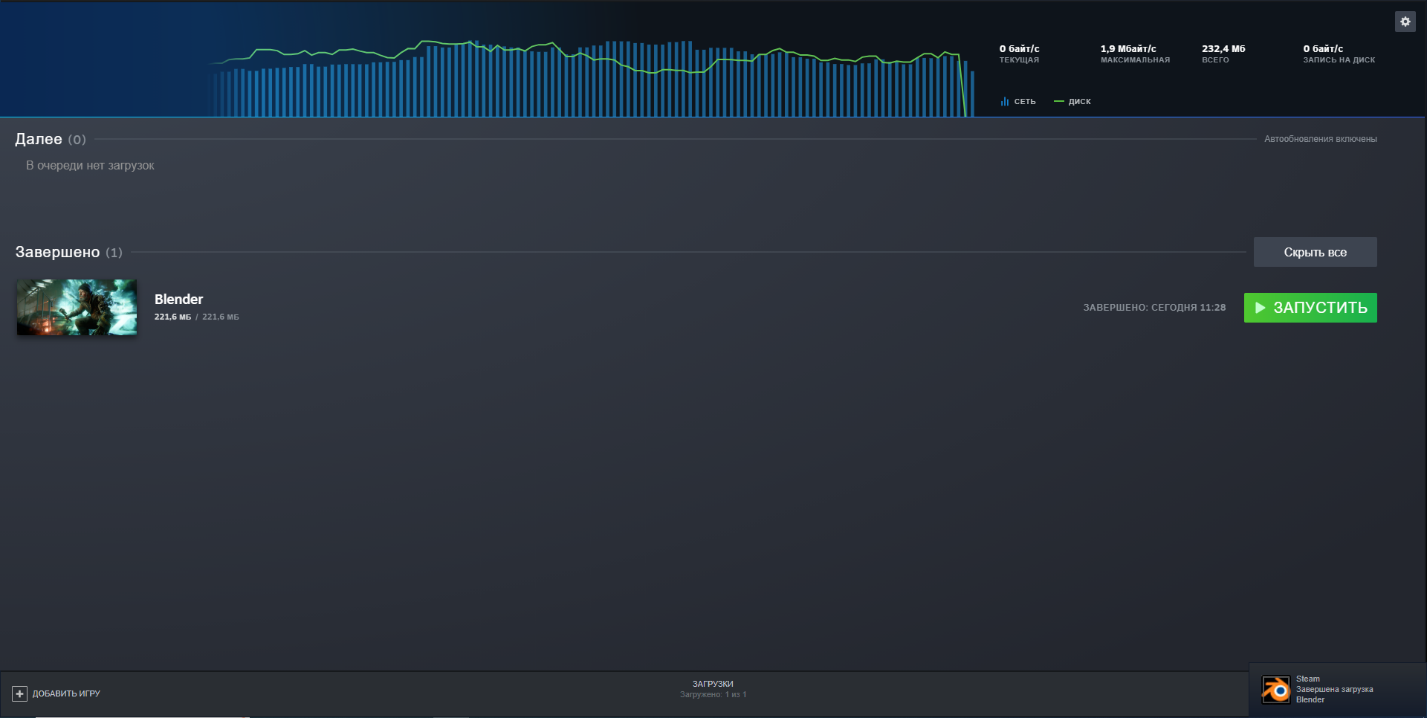
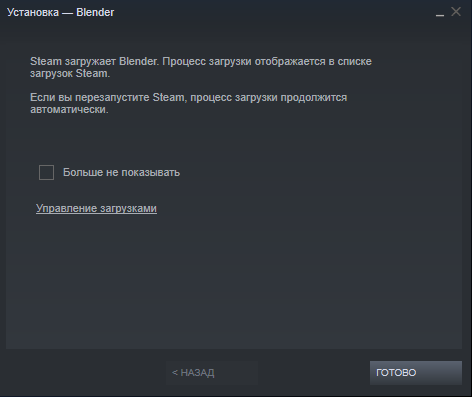
Рисунок 4 – Завершение установки  
  
В результате наших действий приложение успешно загружено, установлено и готов к работе.

Рисунок 5 – Завершение скачивания

* 1. Настройка отдельных компонентов программного обеспечения компьютерных систем и конфигурирование.

При первом запуске нам предлагают выбрать язык, сочетание клавиш, кнопку выделения, действие для пробела и тему.

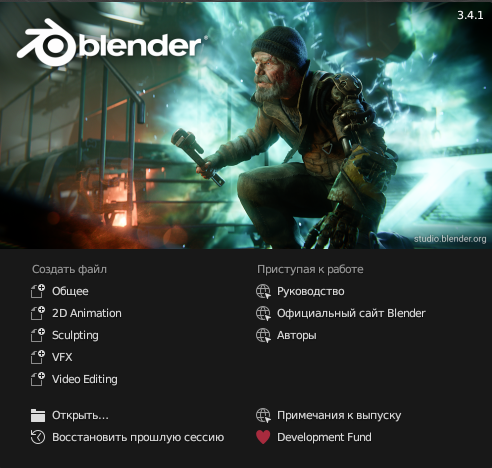
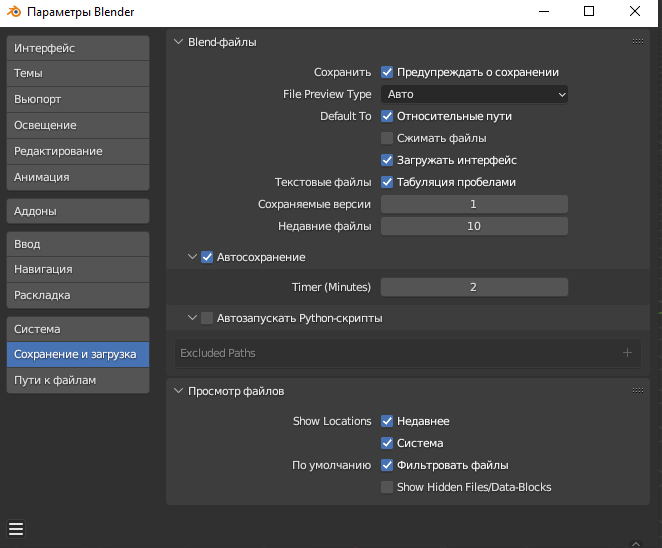
Следом создать файл или выбрать уже готовый:

Рисунок 6

Далее нажимаем «Правка»-«Настройки», у нас открывается окно «Параметры Blender», в котором мы выбираем «Сохранение и загрузка», ищем «Автосохранение» и ставим «Timer» на 2, так же в «Blend-файлы» ищем «Сохраненные версии» и ставим на 32 (В случае экстренного завершения работы программы – вы сможете найти последнюю версию автосохраненной работы в корневой папке):

Рисунок 7  
  
Так же в этом окне есть меню «Ввод», «Навигация» и «Раскладка», в которых вы можете настроить клавиши, способы передвижения в пространстве и т.п.

2 Проблемы совместимости и методы их устранения.

2.1 Тестирование на совместимость в безопасном режиме.

Безопасный режим — это режим запуска системы Windows, предназначенный для устранения неполадок, в котором используется ограниченный набор служб и компонентов. Загружаются только базовые файлы и драйверы, необходимые для запуска Windows. В углах экрана отображаются слова Безопасный режим, соответствующие используемому режиму Windows.

Запускаем Blender в безопасном режиме:

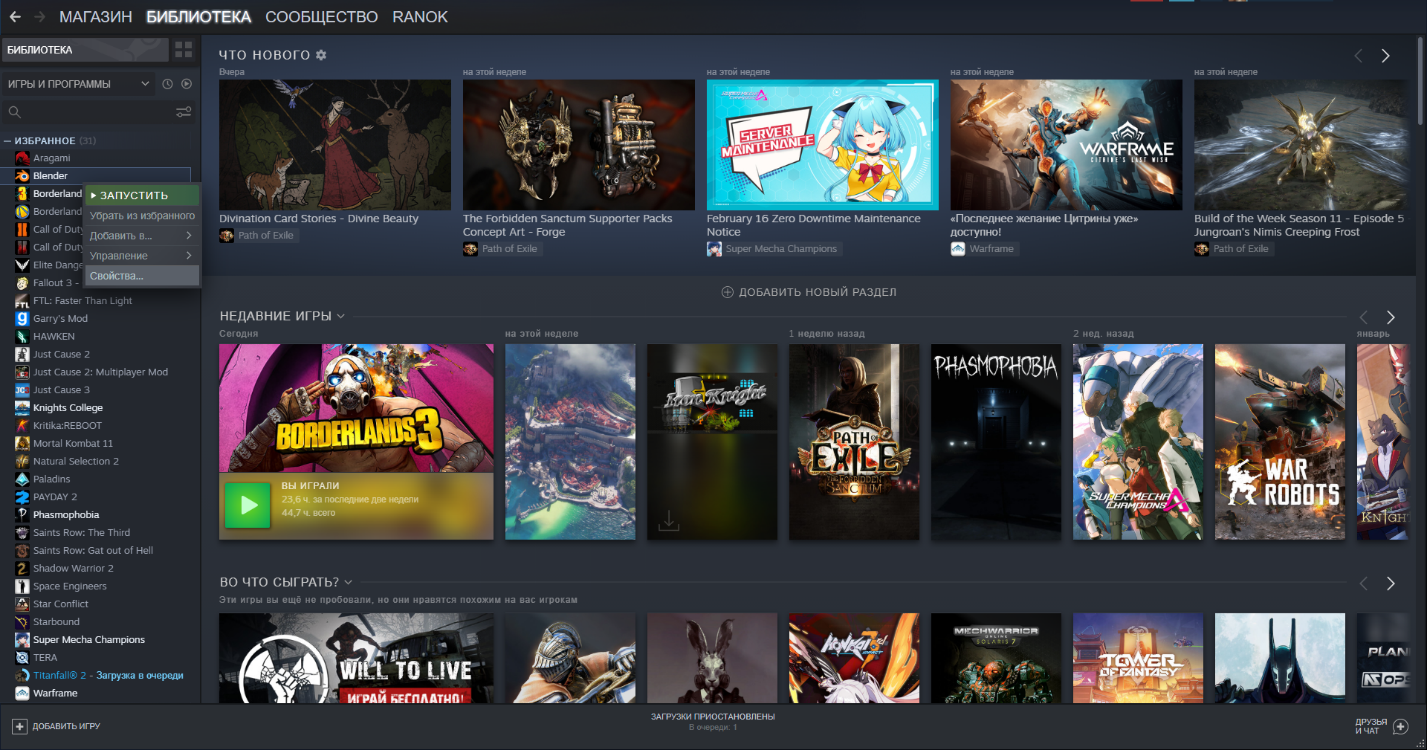
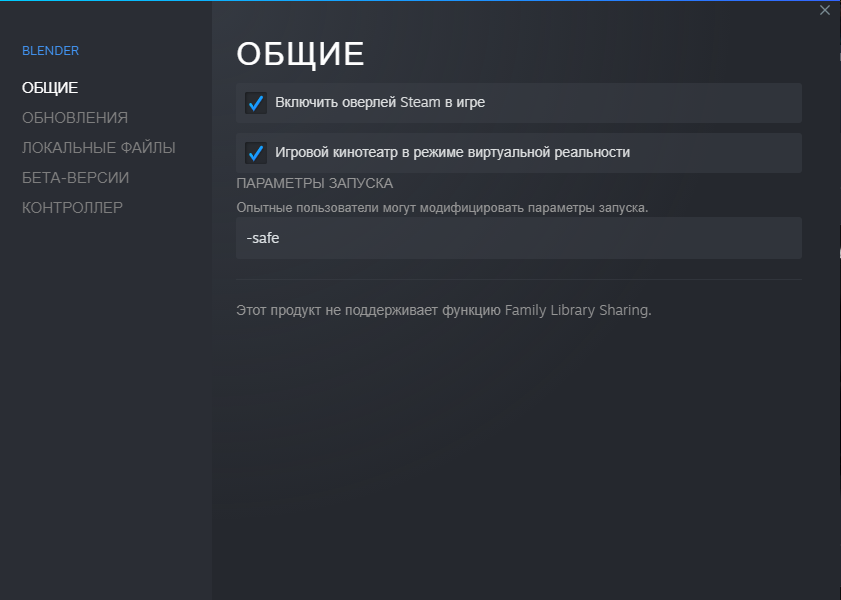
1. Откройте Steam.
2. Перейдите в библиотеку.
3. Выберите Blender.
4. Щелкните правой кнопкой мыши и выберите в контекстном меню «Свойства».

Рисунок 8

1. Перейдите на вкладку «Общие».
2. В строке «Модифицировать параметры запуска» ввести -safe.
3. Закройте окно.
4. Откройте Blender, и он запустится в безопасном режиме.

Рисунок 9

2.2 Инструменты повышения производительности программного обеспечения.

Наиболее эффективные методы повышения производительности:

1. Перенос ПО на более быстрые диск, а именно на SSD, который увеличивает производительность ПО за счет более быстрой обработки информации.
2. Очистка процессора и ОЗУ от ненужных процессов.
3. Очистка кэша процессора и его перераспределение.

2.3 Средства диагностики оборудования.

Диспетчер задач.

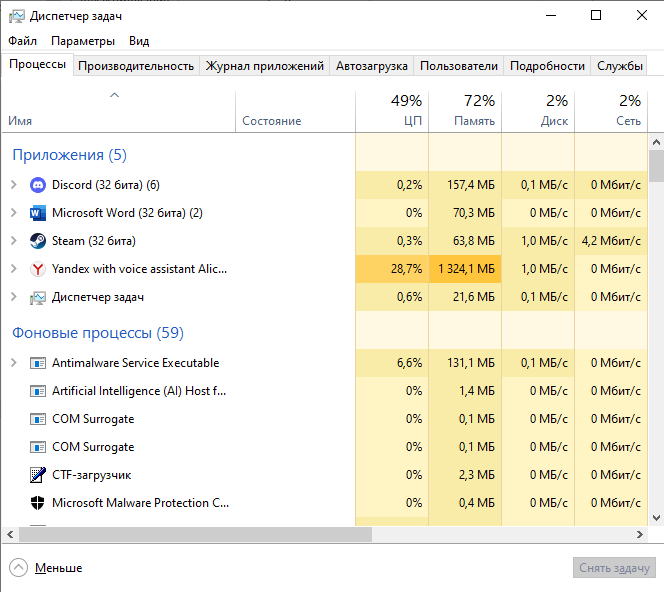
Диспетчер задач (англ. "task manager") – компонент, который входит в состав Windows и позволяет управлять запущенными процессами в режиме реального времени. В новых версиях Диспетчера было добавлено много дополнительных полезных инструментов, которые позволяют производить мониторинг и получать больше данных о работающих задачах в фоне и автозагрузке.

Рисунок 10

CPU-Z

Это бесплатная прикладная программа для отображения технической информации о компьютере пользователя. Программа определяет технические характеристики центрального процессора, материнской платы и BIOS, оперативной памяти, видеокарты, кроме жёсткого диска.

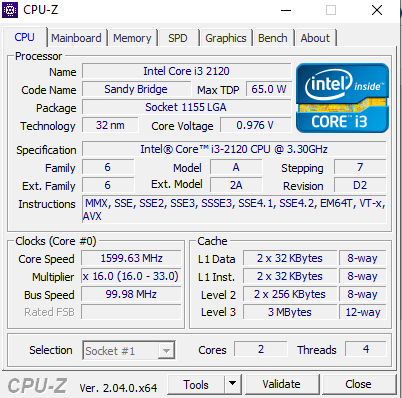


Рисунок 11

2.4 Измерение и анализ эксплуатационных характеристик качества программного обеспечения.

Для объективной оценки качества продукции, ее свойства необходимо охарактеризовать количественно и качественно.

**Качественные характеристики** – это соответствие ПО современному направлению моды, дизайну, цвету и т.д.

**Количественная характеристика** одного или нескольких свойств ПО, составляющих его качество, рассматриваемая применительно к определенным условиям его создания и эксплуатации или потребления (например, безотказность работы, трудоемкость, себестоимость, и т.п.), называется показателем качества продукции (ПКП).

Выбор показателей качества устанавливает перечень наименований количественных характеристик свойств продукции, входящих в состав ее качества и обеспечивающих оценку уровня качества продукции.

Выбраны несколько характеристик качества, на основе которых будет проанализировано ПО:

Blender довольно не удобная программа для новичков, из-за большого количества функций и возможностей.

Blender весьма не стабильно, потому что во время создания больших проектов необходимо больше ресурсов пк.

Blender многофункциональное ПО, в котором можно создавать 2D анимацию, 3D анимацию, 3D модели, скульптуру, редактировать видео и т.п.

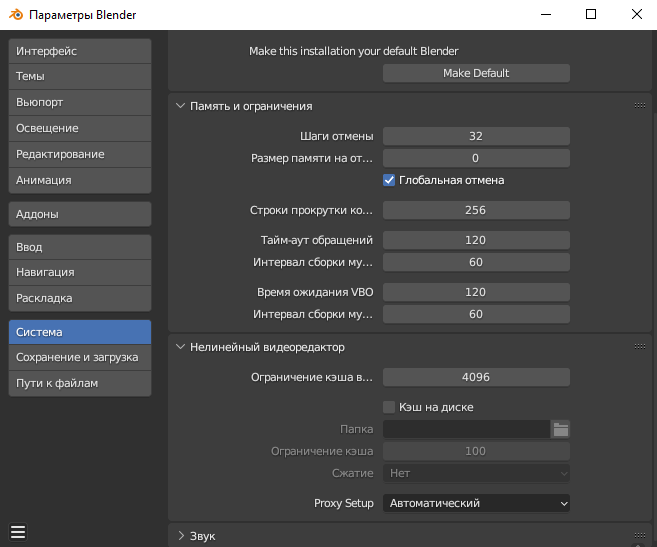
2.5 Оптимизация и модификация ПО.

Настройка и расположение кэша в Blender

Каждый раз, когда вы работаете над проектом, Blender может создавать их кэшированные версии, чтобы иметь возможность быстрого доступа и чтения их с вашего диска.

Со временем эти файлы кэша могут не только заполнить ваше пространство на диске, но и замедлить работу вашего диска и проекта. Вот почему очень важно управлять размером и расположением кэша.

Если у вас есть внутренний твердотельный накопитель, вы можете установить кэш-память на локальный диск, чтобы ускорить работу. Для этого перейдите в «Правка»-«Настройки»-«Система»-«Нелинейный видеоредактор»-«Папка» и нажмите на папку, чтобы изменить место положения кэша.

Рисунок 12

2.6 Разработка руководства оператора

Руководство оператора Blender:

1. Назначение ПО
2. Условия выполнения программы
3. Выполнение программы
4. Сообщения оператору

Примечание:

Руководство писалось с учетом ГОСТ 19.505-79.

3 Этапы сопровождения и обслуживания ПО

3.1 Разработка технического задания на сопровождение

|  |  |  |
| --- | --- | --- |
| Управленческий вопрос | Описание проблемы | Возможное решение |
| Согласование с организационными целями | возврат инвестиций от деятельности по сопровождению программного обеспечения | Программа должна быть универсальной и работать в долгой перспективе, также она должна обеспечить клиента всеми нужными функциями |
| Проблемы кадрового обеспечения | Нужен классифицированный персонал работников, который будет сопровождать ПО | Нужно нанять профессионалов, по сопровождению, которые понимаю свою работу и будут её выполнять верно. Для определения можно использовать тесты, такой вариант подходит если у команды достаточно средств для найма персонала. Либо есть второй вариант обучить новую команду сопровождения, минус этого варианта то-что это затратит больше времени |
| Процесс | Нужно найти ПО, которое поможет в сопровождение | Одна из наиболее детально проработанных и распространенных (на уровне стандарта de facto) процессных моделей, изначально созданных с ориентацией на программное обеспечение – CMMI. Уделяется особое внимание процессам сопровождения. |
| Организационные аспекты сопровождения | Нужно найти организацию для сопровождения | Организация занимающаяся сопровождением |
| Аутсорсинг | Нужно найти корпорацию, которая поможет в сопровождения ПО | Про анализировав все данные было решено передать часть работы компании-исполнителю. В их обязанности входят программные системы |

3.2 Обеспечение качества компьютерных систем в процессе эксплуатации

Обеспечение качества компьютерных систем в процессе эксплуатации является полноценным, если:

* Нужное ПО установлено
* Обоснован вариант конфигурации
* Обеспечен доступ различным категориям пользователей
* Обеспечена совместимость компонента с ранее установленными ПП
* Проконтролировано качество функционирования с помощью встроенных средств

3.3 Разработка сопровождающая документации

Необходимая сопровождающая документация:

* требуется документация разработки и документация продукции для выполнения задач, связанных с обязанностями по обеспечению качества программного обеспечения;
* документация, требующаяся операторам (см.п.2.6), пользователям, руководителям и другим заинтересованным лицам для того, чтобы понимать и использовать программную продукцию;
* сопровождающим программистам требуется детальное описание программного обеспечения, такое, чтобы они могли локализовать и корректировать ошибки и модернизировать или изменять программное обеспечение соответствующим образом;
* документация, требуемая в качестве исторической справки по проекту. Данная документация может помочь в переносе и переводе программного обеспечения в новое окружение.

4 Методы и средства защиты компьютерных систем

4.1 Тестирование на ошибки ввода/вывода данных

При работе в Blender может случиться:

* Ошибка недостатка памяти
* Ошибка недостатка ресурсов ПК

Способы решения:

* Очистите ваш ПК от ненужных файлов
* Используйте меньше эффектов или дополнительных деталей

4.2 Тестирование интерфейса

Этапы тестирования интерфейса для Blender:

* Элементы навигации.

Все кнопки навигации на странице работают и перенаправляют пользователей на нужную страницу или экран.

* Подсказки ввода

В выпадающем меню с сотнями элементов при вводе первой буквы остаются только те элементы, которые начинаются с этой буквы.

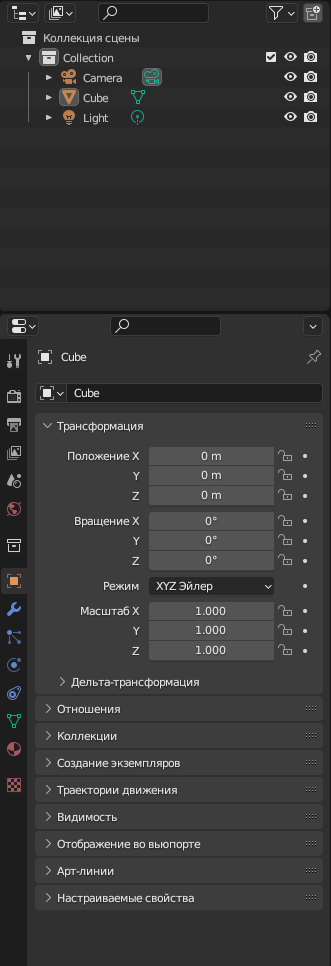


Рисунок 13

* Панель с элементами.

Отображает все элементы, с которыми вы работаете.

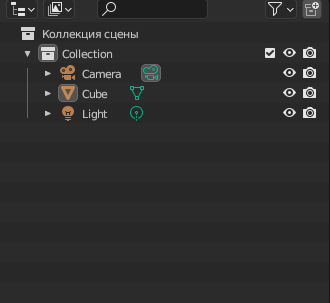


Рисунок 14

4.3 Тестирование контента

---

4.4 Анализ рисков. Выявление первичных и вторичных ошибок.

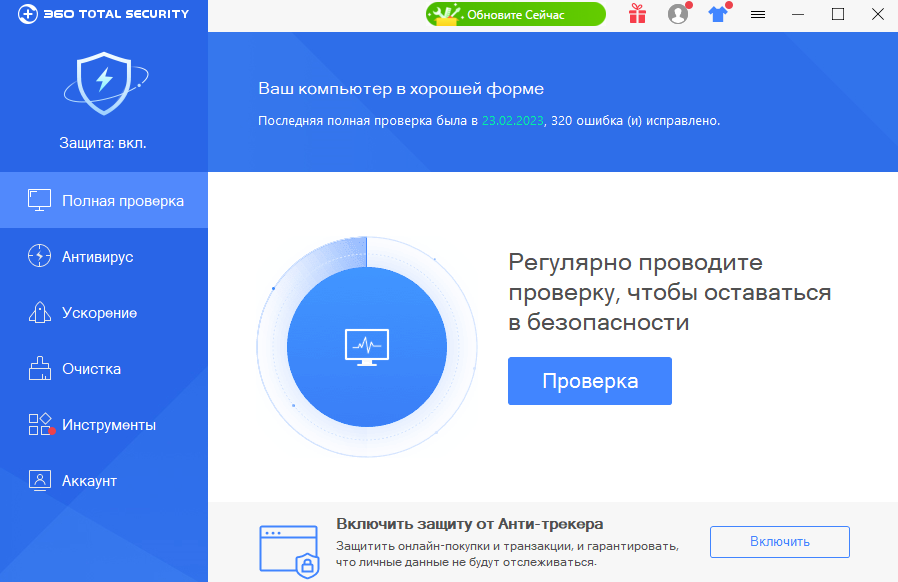
Первичные ошибки, как правило, бывают следующими:

* Технологические ошибки
* Алгоритмические ошибки, связанные с некорректной постановкой задач
* Системные ошибки

Вторичные – последствия и результаты некоторых дефектов

* Сбои
* Ординарные отказы
* Катастрофические отказы
* Перегрузки
* Ошибки тестирования
* Ошибка выявлена и забыта

4.5 Обнаружение вируса и устранение последствий его влияния.

Рисунок 15  
  
360 Total Security – основанная на двух движках антивирусная программа и комплекс утилит для оптимизации и очистки системы. После установки и при первом запуске программа предложит провести сканирование системы. Она проверит ОС на вирусы, «мусорные» фалы и безопасность сети.

4.6 Тестирование защиты программного обеспечения

У Blender нет защитных мер

4.7 Работа с реестром. Работа с программой восстановления файлов и очистки дисков.

Системный реестр – обширная база данных, в которой хранится большинство настроек установленных программ и самой Windows

Необходимость очистки возникает из-за неправильного удаления приложений, сбоев в работе системы и деятельности вирусов. В этих случаях появляются ошибки реестра. Очистка является довольно рискованной операцией. Даже самые надёжные программы не всегда могут произвести достаточно глубокий анализ, чтобы определить, какие изменения повлечёт удаление той или иной записи. Поэтому рекомендуется всегда сохранять бэкапы (резервные копии) удалённых данных.

Работать с реестром можно с помощью сторонних программ, например, Auslogics BoostSpeed

С помощью данной утилиты можно:

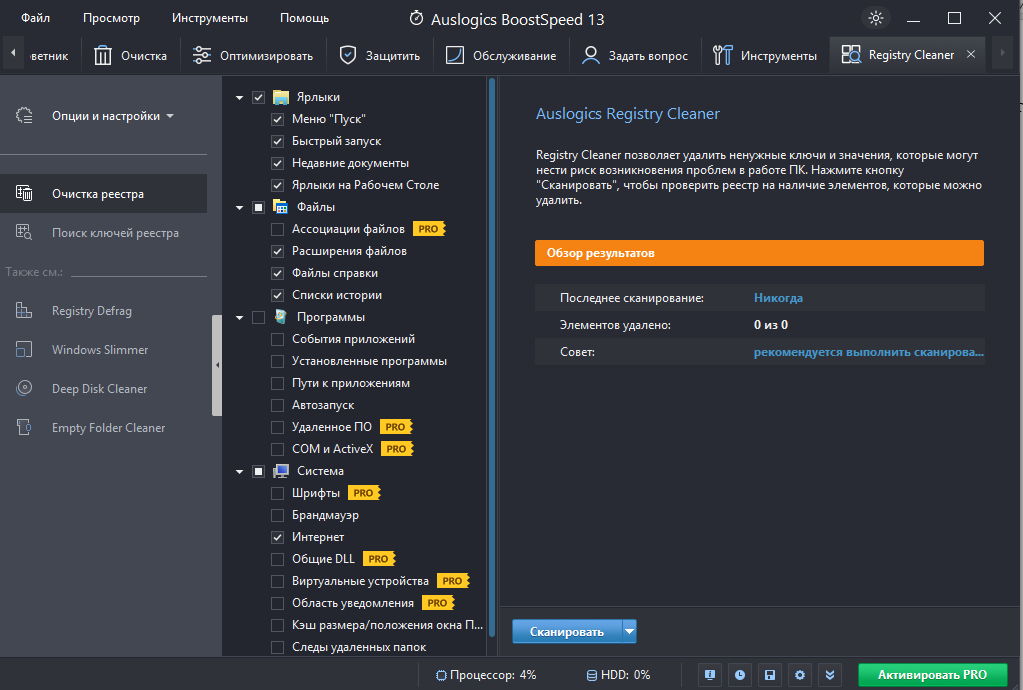
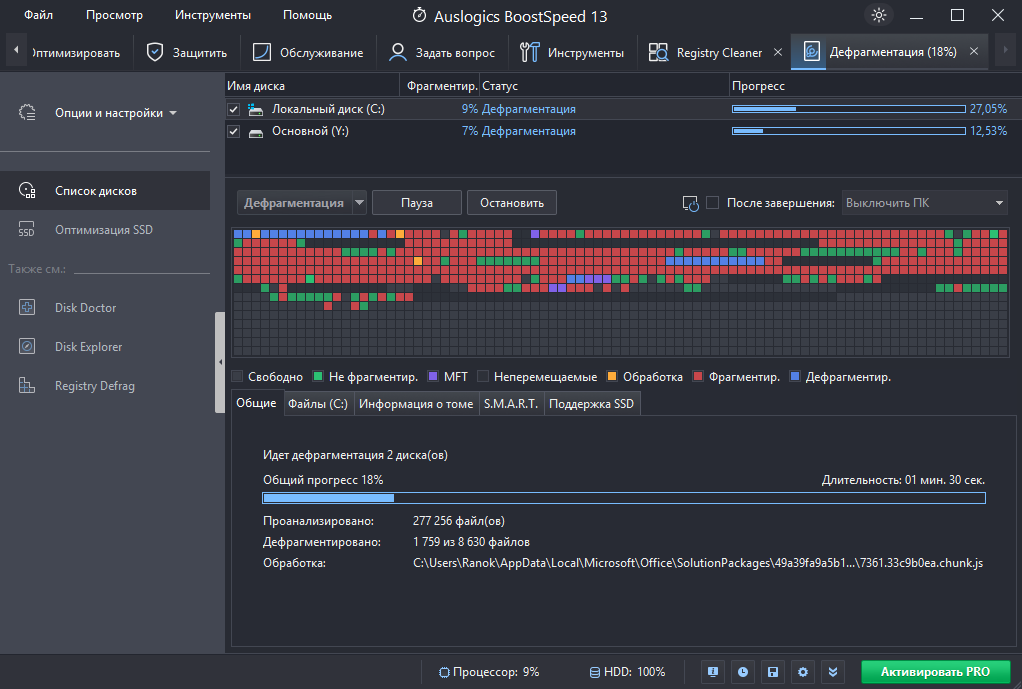
* Очистить реестр

Рисунок 16

* Сделать дефрагментацию диска

Рисунок 17

5 Индивидуальное задание

5.1 Практические работы

Практическая работа №1 «Внедрение программного обеспечения»

Внедрение программного обеспечения — процесс настройки программного обеспечения под определенные условия использования, а также обучения работе с программным продуктом.

Внедрение программного продукта состоялось в том случае, если программный продукт выполняет поставленную задачу, а сотрудники компании полностью перешли на работу с новым продуктом.

Результатом проекта внедрения должен быть прозрачный, четко регламентированный, документированный и автоматизированный процесс разработки и сопровождения.

Основные этапы внедрения программного продукта:

1. Обследование

2. Разработка ТЗ

3. Настройка системы (программного продукта)

4. Тестирование системы

5. Опытная эксплуатация

6. Промышленная эксплуатация

Эффекты от внедрения можно разделить на две большие категории: эффект, получаемый на стратегическом уровне, т.е. на уровне бизнеса, и тактический эффект, который получат все участники процесса.

1. Стратегический эффект

* обеспечение прозрачности и измеряемости достижения стратегических целей;
* снижение вероятности проявления рисков в сфере информационных технологий;
* повышение рентабельности IT-услуг за счет снижения сроков проведения проекта, снижения издержек поддержки и т.д.;
* повышение инвестиционной привлекательности IT-проектов;
* повышение доверия бизнеса к IT за счет качественно обработанных запросов IT-отделами от бизнеса.

2. Тактический эффект

* уменьшение сроков и снижение стоимости обработки новых запросов;
* улучшение качества IT-услуг – это вовремя выполненные работы с заранее определенным качеством;
* увеличение эффективности используемых ресурсов – хорошо организованный и формализованный процесс позволяет четко определить роли и ответственности каждого участника этого процесса;
* более четкое и реалистичное планирование;
* значительное уменьшение времени на принятие решения;
* снижение влияния человеческого фактора.

Сопровожде́ние (поддержка) программного обеспечения — процесс улучшения, оптимизации и устранения дефектов программного обеспечения (ПО) после передачи в эксплуатацию.

Сопровождение ПО — это одна из фаз жизненного цикла программного обеспечения, следующая за фазой передачи ПО в эксплуатацию.

Определение процесса сопровождения:

* + SWEBOK: Сопровождение ПО – вся совокупность деятельности, необходимой для обеспечения эффективной поддержки программных систем.
  + IEEE 1219 (Standard for Software Maintenance): Сопровождение ПО – определяется как модификация программного продукта после передачи в эксплуатацию для устранения сбоев, улучшения показателей производительности и/или других характеристик (атрибутов) продукта, или адаптации продукта для использования в модифицированном окружении.
  + ГОСТ Р ИСО/МЭК 12207: Сопровождение – процесс модификации программного продукта в части его кода и документации для решения возникающих проблем при эксплуатации или реализации потребностей в улучшениях тех или иных характеристик продукта.

Сопровождение поддерживает функционирование программного продукта на протяжении всего операционного жизненного цикла, то есть периода его эксплуатации.

В процессе сопровождения фиксируются и отслеживаются запросы на модификацию (также называемые “запросами на изменения” – change requests, в частности, в контексте конфигурационного управления), оценивается влияние предлагаемых изменений, производится модификация кода и других активов (артефактов) продукта, проводится необходимое тестирование и, наконец, выпускается обновленная версия продукта. Кроме того, проводится обучение пользователей и обеспечивается их ежедневная поддержка при работе с текущей версией продукта.

Практическая работа №2 «Техническая поддержка ПО»

Образец договора технической поддержки программного обеспечения.

1. Термины договора на сопровождение программ

1.1. Продукты – предоставленное по данному договору № 135 от «13» мая 2022 г. программное обеспечение.

1.2. Техническая поддержка - оказываемые услуги по настройке, обслуживанию, адаптации и модификации продуктов или устранению имеющихся в них ошибок, а также предоставлению обновлений и дополнительных программных модулей, иные действия предусмотренные в разделе 2 настоящего Договора.

1.3. Ошибка - дефект в коде Продукта, в результате которого данный Продукт не способен работать в соответствии функциональными возможностями, указанными в предоставленной на него технической документации за исключением случаев:

(1) нарушения Заказчиком правил эксплуатации продуктов в соответствии с требованиями предоставленной на них технической документации;

(2) использования Продуктов на оборудовании или совместно с программным обеспечением, которые не были рекомендованы Исполнителем.

2. Предмет договора техподдержки ПО

2.1. Исполнитель обязуется оказывать по заявкам Заказчика услуги по Технической поддержке, а Заказчик принимать и оплачивать оказываемые Исполнителем услуги.

2.2. Стороны согласовали возможность оказания услуг по Технической поддержке в следующем объеме и составе:

1. Установка Продуктов на оборудование Заказчика;
2. Настройка Продуктов на оборудовании Заказчика, включая их адаптацию;
3. Модификация Продуктов по отдельному заказу;
4. Предоставление выпускаемых обновлений Продуктов;
5. Устранение ошибок в Продуктах;
6. Консультация по порядку использования Продуктов.

2.3. Услуги по Договору оказываются в соответствии с соглашением об уровне услуг(SLA), являющимся неотъемлемой частью настоящего Договора.

Практическая работа №3 «Защита ПО»

Методы доказательства правильности программ могут быть применены для разработки ПО при существенных ограничениях на размеры и сложность создаваемых программ. Поэтому в частных случаях они могут оказаться более эффективными, чем другие известные методы анализа программ, которые исследуются в следующих разделах данной работы.

Методы, используемые для анализа и оценки безопасности ПО, разделяют на две категории: контрольно-испытательные и логико-аналитические. В основу данного разделения положены принципиальные различия в точке зрения на исследуемый объект (программу).

Контрольно-испытательные методы анализа рассматривают РПС через призму фиксации факта нарушения безопасного состояния системы, а логико-аналитические - через призму доказательства наличия отношения эквивалентности между моделью исследуемой программы и моделью РПС.

Контрольно-испытательные методы делятся на те, в которых контролируется процесс выполнения программы и те, в которых отслеживаются изменения в операционной среде, к которым приводит запуск программы.

При проведении анализа безопасности с помощью логико-аналитических методов строится модель программы и формально доказывается эквивалентность модели исследуемой программы и модели РПС. В простейшем случае в качестве модели

В целом полный процесс анализа ПО включает в себя три вида анализа:

1. Лексический;
2. Синтаксический;
3. Сематический.

Каждый из видов анализа представляет собой законченное исследование программ согласно своей специализации.

Результаты исследования могут иметь как самостоятельное значение, так и коррелироваться с результатами полного процесса анализа.

Лексический анализ предполагает поиск распознавания и классификацию различных лексем объекта исследования (программа), представленного в исполняемых кодах. При этом лексемами являются сигнатуры. В данном случае осуществляется поиск сигнатур следующих классов:

1. сигнатуры вирусов;
2. сигнатуры элементов РПС;
3. сигнатуры (лексемы) "подозрительных функций";
4. сигнатуры штатных процедур использования системных ресурсов и внешних устройств.

Поиск лексем (сигнатур) реализуется с помощью специальных программ-сканеров.

Синтаксический анализ предполагает поиск, распознавание и классификацию синтаксических структур РПС, а также построение структурно-алгоритмической модели самой программы.

Решение задач поиска и распознавания синтаксических структур РПС имеет самостоятельное значение для верификационного анализа программ, поскольку позволяет осуществлять поиск элементов РПС, не имеющих сигнатуры. Структурно-алгоритмическая модель программы необходима для реализации следующего вида анализа - семантического.

Сематический анализ предполагает исследование программы изучения смысла составляющих ее функций (процедур) в аспекте операционной среды компьютерной системы. В отличие от предыдущих видов анализа, основанных на статическом исследовании, семантический анализ нацелен на изучение динамики программы - ее взаимодействия с окружающей средой. Процесс исследования осуществляется в виртуальной операционной среде с полным контролем действий программы и отслеживанием алгоритма ее работы по структурно-алгоритмической модели.

Сематический анализ является наиболее эффективным видом анализа, но и самым трудоемким. По этой причине методика сочетает в себе три перечисленных выше анализа. Выработанные критерии позволяют разумно сочетать различные виды анализа, существенно сокращая время исследования, не снижая его качества.

Практическая работа №4

|  |  |
| --- | --- |
| Тип комплектующих | Модель комплектующих |
| Процессор | Intel i3-10400F |
| Оперативная память | 6Gb DDR 4 |
| Видеокарта | NVIDIA GeForce GTX 660 Ti |
| Звуковая плата | Интегрированный контроллер |
| Сетевая карта | Интегрированный контроллер |
| Жёсткий диск | 1Tb |
| Дисковод для гибких дисков | Отсутствует |
| Оптический привод | Отсутствует |
| Клавиатура | SVEN GS4300 |
| Мышь | SVEN GS4300 |
| Монитор | LG 22EN33 |
| Корпус | Eterno |
| Блок питания | 500W |
| Источник бесперебойного питания | Отсутствует |

5.2 Методы верификации программного обеспечения

К методам верификации программного обеспечения относятся статический анализ программ (проведение инспекций и экспертиз, анализ архитектуры программного обеспечения, проверка моделей и дедуктивный анализ корректности программного обеспечения), а также методы динамической верификации – тестирование и верификационный мониторинг.

5.3

Инсталлятор - это "обычная" программа, основные функции которой - Установка (Инсталляция), Обновление и Удаление (Деинсталляция) программного обеспечения.

Являясь обычной программой, инсталлятор обладает рядом особенностей, среди которых стоит отметить следующие:

* глубокое взаимодействие с операционной системой и зависимость от неё (файловая система, реестр, сервисы и библиотеки)
* совместимость как родных, так и сторонних библиотек, компонент или драйверов, с разными платформами
* удобство использования: интуитивно понятный интерфейс, навигация, сообщения и подсказки
* дизайн и стиль инсталляционного приложения
* совместимость пользовательских настроек и документов в разных версиях приложения

И многое другое

Тестирование мастера установки (Installation Wizard)

План тестирования инсталляционного визарда:

Определить все пути от начала до конца, и затем расставить приоритеты для каждого из них. Это поможет нам избежать излишних затрат и усилий при прохождении низкоприоритеных путей.

Забудьте про GUI. Постарайтесь описать тест-кейзы без привязки к интерфейсным элементам. К примеру, GUI контролы checkbox/radiobutton или меню из двух пунктов это просто выбор между true и false, важно то, на что он влияет в конечном счете.

Если по результатам прохождения визарда получается какой либо проперти файл (файл, описывающий свойства в виде списка: свойство=значение), который потом передается дальше в процедуру экспорта. В этом случае можно разделить проверки на два этапа - первый, создавать (генерировать) такие проперти файлы и проверять, что экспорт работает правильно. Второй - проверять, что через GUI получаются правильные проперти файлы.

Не забудьте заняться таким рутинным видом тестирования визардов, как ходить туда-обратно по страницам:

* ничего не меняя, все ответы должны сохраняться;
* меняя что-либо на предыдущей странице, на следующей должно произойти адекватное изменение либо сброс ответов.

Убедитесь, что визард адекватно реагирует на неправильные ответы и не дает ходить дальше.

Кнопка Cancel (Close) должна работать всегда и на всех страницах визарда.

Создайте для каждого из возможных путей мастера установки шаблонный результат (в идеале, сделайте их несколько - для разных входных данных). Затем, по возможности, автоматизировано или вручную сравнивайте полученный результат с шаблонным.

Выделите те опции, которые не влияют ни на какие другие, и на которые другие не оказывают влияния. Работу этих опций можно будет тестировать изолированно от других.

Сравнительный анализ архиваторов

Архиватор — программа, осуществляющая сжатие и/или упаковку одного и более файлов в архив или серию архивов, для удобства переноса или хранения, а также распаковку архивов. На сегодняшний день количество архиваторов насчитывается около 50… Некоторые из них устарели и практически не используются. Каждый архиватор обладает своими достоинствами и недостатками, такими как скорость сжатия и степень сжатия. Эти характеристики — обратно зависимые величины. То есть, чем больше скорость сжатия, тем меньше степень сжатия, и наоборот. В настоящее время архивирование (сжатие) данных является трудоемкой задачей. Дело в том, что большинство пользователей или владельцев (авторов) информации хранят различную информацию в сжатом виде (в архивах), что уменьшает размер и в некоторой степени защищает ее. Защита подразумевает собой защиту от некоторых вирусов, т.к. сжатие данных – это есть кодирование информации в некоторой степени. А само сжатие данных обычно происходит значительно медленнее, чем обратная операция (разархивирование).

Заключение

В данном отчете мы узнали достоинства и возможности данного ПО. Научились его настраивать, включать безопасный режим, пользоваться антивирусом, работать с реестром. Узнали про средства диагностики оборудования.

Список использованных источников:

1) <https://ip-calculator.ru/blog/ask/kak-zapuskat-igry-v-bezopasnom-rezhime-v-steam-na-windows-10/>

2)<https://ru.wikipedia.org/wiki/Blender>

3) <https://blender-ru.com/>

4) <https://normativ.kontur.ru/document?moduleId=1&documentId=124814>

5) <https://app.emaze.com/@AOTFTQCFR>

6) <http://studentpmr.ru/?p=284>

7) <https://www.swrit.ru/doc/espd/19.505-79.pdf>

8) <https://www.cpuid.com/softwares/cpu-z.html>

9) <https://www.360totalsecurity.com/ru/>

10) <https://www.auslogics.com/ru/software/boost-speed/>