АКТ(ф) СПбГУТ

ОТЧЕТ

по лабораторным и практическим работам

Обеспечение качества функционирования компьютерных систем

Студент ИСПП-21	18.12.2025	С.А.Миклякова
Преподаватель	18.12.2025	Р.В.Садовский

Архангельск 2025

Лабораторная работа №1
Оценка качества программного обеспечения

1 Цель работы

1.1 Познакомиться с методами и инструментами оценки качества программного обеспечения;

1.2 Научиться применять основные метрики и критерии для оценки функциональности, производительности, безопасности и удобства использования программных продуктов.

2 Ход работы

2.1 Рассмотрели программный продукт, предоставленный преподавателем, по применимым внешним метрикам и характеристикам качества ПО;

2.2 Оценили характеристики и метрики качества ПО и занесли значения в таблицу.

3 Ответы на контрольные вопросы

3.1 ГОСТ ИСО/МЭК 9126-01 определяет набор характеристик качества ПО, которые делятся на следующие основные категории:

- функциональность: способность ПО выполнять заданные функции;
- надежность: способность ПО поддерживать заданный уровень работоспособности в течение определенного времени;
- удобство использования: возможность пользователя эффективно, удобно и просто взаимодействовать с ПО;
- эффективность: производительность и использование ресурсов;
- модифицируемость: простота внесения изменений в ПО;

- тестируемость: возможность проведения тестирования и обнаружения ошибок.

3.2 Основные методы оценки удобства использования включают:

- экспертные оценки: привлечённые эксперты проводят анализ интерфейса и взаимодействия, выявляют потенциальные проблемы;
- пользовательские тесты: реальные или предполагаемые пользователи выполняют задачи с программой, после чего собирается обратная связь по удобству, эффективности и проблемам;
- анкетирование и опросы: использование специальных опросных листов для оценки восприятия удобства, сложности и эффективности использования ПО;
- метод наблюдения: наблюдение за реальными пользователями при работе с программой для определения проблемных областей;
- методы автоматизированного анализа: использование специальных инструментов для анализа интерфейса.

4 Вывод

4.1 В ходе лабораторной работы изучены методы и инструменты оценки качества программного обеспечения;

4.2 В ходе лабораторной работы изучены получены навыки применять основные метрики и критерии для оценки функциональности, производительности, безопасности и удобства использования программных продуктов.

Лабораторная работа №2
Исследование уязвимостей ПО и методов их устранения

1 Цель работы

1.1 Изучить основные угрозы безопасности веб-приложений, такие как SQL-инъекции, XSS-инъекции, а также методы защиты от них;

1.2 Научиться применять защитные меры в разработке безопасных веб-приложений.

2 Ход работы

2.1 Реализовали и запустили приложение, подверженное уязвимостям;

2.2 Протестировали SQL-инъекции;

2.3 Протестировали XSS-инъекции;

2.4 Протестировать уязвимости авторизации;

2.5 Исправили код с уязвимостями.

3 Ответы на контрольные вопросы

3.1 Для защиты от SQL-инъекций используются параметризованные запросы хранимые процедуры и валидация ввода.

3.2 Защита от XSS-инъекций начинается с контекстного экранирования вывода. Content Security Policy блокирует выполнение скриптов из неавторизованных источников через HTTP-заголовки.

4 Вывод

4.1 В ходе лабораторной работы изучены основные угрозы безопасности веб-приложений, такие как SQL-инъекции, XSS-инъекции, а также методы защиты от них;

4.2 В ходе лабораторной работы получены навыки применять защитные меры в разработке безопасных веб-приложений.

Лабораторная работа №3
Применение статических анализаторов программного кода

1 Цель работы

1.1 Изучить процесс использования статических анализаторов кода для повышения качества программного обеспечения.

2 Ход работы

2.1 Установили PVS-studio;

2.2 Создали новый проект C#;

2.3 Проанализировали код.

3 Ответы на контрольные вопросы

3.1 Статический анализ кода – это процесс автоматизированной проверки исходного кода программы без его фактического выполнения. Его основные преимущества включают раннее обнаружение проблем, повышение качества и читаемости кода.

3.2 Статический анализ работает с исходным кодом статично, без выполнения программы, и обнаруживает широкий спектр проблем вроде нарушений стиля или потенциальных багов. Динамический анализ требует запуска приложения с тестовыми данными, фокусируясь на runtime-ошибках, производительности и реальном поведении, но охватывает лишь исполняемые пути.

3.3 SonarLint в C# обнаруживает обращение к свойству null-объекта, выход за границы массива, вызов метода с null-параметром без валидации, дублированный код, слишком длинные методы, SQL и XSS-инъекции, утечку данных, неиспользуемые using-директивы и переменные.

3.4 Статические анализаторы пропускают runtime-ошибки, возникающие только при выполнении с реальными данными, такие как деление на ноль с динамическими значениями или NullReferenceException в зависимости от внешнего ввода.

4 Вывод

4.1 В ходе лабораторной работы изучен процесс использования статических анализаторов кода для повышения качества программного обеспечения.

Лабораторная работа №4
Применение оперативных методов повышения надежности ПО

1 Цель работы

1.1 Изучить процесс применения оперативных методов повышения надежности ПО.

2 Ход работы

2.1 Реализовали временную избыточность для выполнения запроса к серверу в условиях ненадёжного сетевого соединения;

2.2 Реализовали программную избыточность для анализа аномалий в измерении температуры;

2.3 Реализовали информационную избыточность методом зеркалирования данных и проверки чек-сумм.

3 Ответы на контрольные вопросы

3.1 Временная избыточность использует повторное выполнение операций или контрольные циклы для обнаружения и исправления ошибок без дублирования аппаратных ресурсов. Ее преимущества включают низкие затраты на реализацию и простоту интеграции в существующие системы. Недостатки проявляются в увеличении времени выполнения задач и зависимости от скорости процессора.

3.2 Программная избыточность применяет дублирующие алгоритмы или модули ПО для резервирования, обеспечивая отказоустойчивость на уровне кода. Преимущества состоят в гибкости настройки и возможности восстановления без аппаратных изменений, а недостатки – в повышенной сложности разработки и риске синхронизации ошибок между копиями.

3.3 Информационная избыточность добавляет контрольные суммы, чередующие коды или парности для проверки целостности данных. Она выигрывает в точном обнаружении повреждений и минимальном влиянии на производительность, однако требует дополнительного объема хранения и может не исправлять ошибки автоматически.

4 Вывод

4.1 В ходе лабораторной работы изучен процесс применения оперативных методов повышения надежности ПО.

Лабораторная работа №5
Анализ рисков и характеристик качества ПО при внедрении

1 Цель работы

1.1 Изучить процесс анализа рисков ПО при разработке и внедрении.

2 Ход работы

2.1 Выполнили анализ рисков при разработке и внедрении ПО;

2.2 Составили таблицу оценки и минимизации рисков внедрения ПО.

3 Ответы на контрольные вопросы

3.1 Технические риски включают возможность появления ошибок и дефектов в ПО, которые могут привести к сбоям или уязвимостям. Важным аспектом является недостаточная производительность или масштабируемость системы, что может ограничить её использование.

3.2 К организационным рискам относятся несогласованность между участниками проекта, плохое управление проектом, отсутствие четких требований и недостаточное взаимодействие команд.

3.3 Экономические риски включают превышение бюджета проекта из-за недооценки сложности, задержек или неожиданных расходов. Также есть риск низкой рентабельности или отсутствия окупаемости инвестиций, особенно если рынок изменится или система окажется невостребованной.

3.4 Юридические риски связаны с возможными нарушениями авторских прав, лицензий и патентов при использовании сторонних компонентов или технологий. Несоблюдение требований законодательства о защите персональных данных, конфиденциальности и информационной безопасности также создает риски юридической ответственности.

4 Вывод

4.1 В ходе лабораторной работы изучен процесс анализа рисков ПО при разработке и внедрении.

Лабораторная работа №6
Использование архитектурных решений для обеспечения качества ПО

1 Цель работы

1.1 Изучить процесс разработки приложений с использованием микросервисной архитектуры;

1.2 Изучить процесс развертывания приложений с использованием микросервисной архитектуры.

2 Ход работы

2.1 Создали виртуальную машину Ubuntu LiveServer;

2.2 Установили Docker;

2.3 Разработали API;

2.4 Разработали API-шлюза;

2.5 Указали настройки Api-шлюза в файле ocelot.json;

2.6 Настроили решение для работы с контейнерами.

3 Ответы на контрольные вопросы

Преимущества микросервисной архитектуры включают в себя:

- масштабируемость: каждый микросервис можно масштабировать независимо, что позволяет повышать производительность системы;
- гибкость технологий: разные микросервисы могут быть реализованы на различных языках программирования или использовать разные технологии;
- надежность: отказ одного микросервиса не обязательно влечет за собой отказ всей системы, что повышает отказоустойчивость;
- легкость обновлений и внедрения новых функций;

- улучшенная поддержка и развитие: команды могут работать над разными микросервисами параллельно, ускоряя разработку и повышая качество;

К недостаткам микросервисной архитектуры относится:

- сложность управления: множество микросервисов требуют сложной инфраструктуры для оркестрации, мониторинга и обработки ошибок;
- проблемы с согласованностью данных: обеспечение целостности данных при распределенной архитектуре сложнее, чем при монолитной;
- сложности в тестировании: интеграционное тестирование и отладка взаимодействия микросервисов может быть сложнее.

4 Вывод

4.1 В ходе лабораторной работы изучен процесс разработки приложений с использованием микросервисной архитектуры;

4.2 В ходе лабораторной работы изучен процесс развертывания приложений с использованием микросервисной архитектуры.

Лабораторная работа №7
Разработка адаптируемого ПО

1 Цель работы

1.1 Изучение процесса разработки ПО с поддержкой внешней интеграции при помощи скриптов.

2 Ход работы

2.1 Реализовали поддержку скриптинга;

2.2 Реализовали поддержку внешних скриптов;

2.3 Добавили в приложение интерфейс для работы с событиями;

2.4 Реализовать методы для автоматизации работы с событиями (CRUD) посредством скриптов;

2.5 Добавить возможность выполнять скрипты из заранее записанных текстовых файлов.

3 Ответы на контрольные вопросы

Способы взаимодействия между приложениями включают использование API, при котором приложения вызывают друг друга через стандартизированные интерфейсы. Также широко применяется обмен сообщениями с помощью брокеров сообщений, что обеспечивает асинхронную работу и высокую масштабируемость.

4 Вывод

4.1 В ходе лабораторной работы изучен процесс разработки ПО с поддержкой внешней интеграции при помощи скриптов.

Лабораторная работа №8
Защита программного обеспечения на уровне кода

1 Цель работы

1.1 Познакомиться с методами защиты исходного кода от декомпиляции и анализа.

2 Ход работы

2.1 Декомпилировали приложение;

2.2 Использовали Obfuscar;

2.3 Исключили код из обфускации;

2.4 Реализовали обфускацию оконного приложения.

3 Ответы на контрольные вопросы

3.1 Обфускация применяется для защиты программного кода от несанкционированного анализа и копирования.

3.2 Обфускация работает путём преобразования исходного кода или машинного кода таким образом, что его функциональность сохраняется, а структурные и логические элементы делают его трудным для восприятия.

4 Вывод

4.1 В ходе лабораторной работы изучены методы защиты исходного кода от декомпиляции и анализа.

Лабораторная работа №9
Исследование правовых аспектов защиты ПО

1 Цель работы

1.1 Познакомиться с методами правовой защиты ПО.

2 Ход работы

2.1 Используя источники сети Интернет заполнили сравнительную таблицу open-source лицензий ПО;

2.2 Используя ресурсы сайта rospatent.gov.ru, подготовили пакет документов для государственной регистрации программы для ЭВМ.

3 Ответы на контрольные вопросы

3.1 Проприетарное программное обеспечение – это программа, права на которую принадлежат одному лицу или организации. Пользователям обычно предоставляется ограниченная лицензия на использование, а исходный код скрыт или недоступен для просмотра. Такой софт отличается от свободного или открытого программного обеспечения строгими ограничениями на копирование, распространение и модификацию.

3.2 Лицензии для ПО разрабатывают правообладатели. Это могут быть компании, организации или авторы программ. Также могут заниматься этим специальные юристы или организации, специализирующиеся на разработке лицензионных условий.

3.3 Авторское право – это юридическая гарантия, которая защищает права создателей оригинальных произведений. Оно предоставляет автору исключительные права на использование, распространение, изменение и

публичное показ своих произведений, а также регулирует возможности других лиц использовать созданное им произведение без разрешения.

4 Вывод

4.1 В ходе лабораторной работы изучены методы правовой защиты ПО.

Лабораторная работа №10
Исследование способов антивирусной защиты компьютерных систем

1 Цель работы

1.1 Познакомиться с методами антивирусной защиты ПО.

2 Ход работы

2.1 Используя сайт VirusTotal.com идентифицировали файлы по их хэшсуммам и определили, являются ли они вредоносными;

2.2 Проверили 5 различных веб-сайтов при помощи VirusTotal;

2.3 Используя API VirusTotal написали консольное приложение для проверки выбранного файла на вирусные сигнатуры.

3 Ответы на контрольные вопросы

3.1 Антивирусные программы делятся по принципу работы (сканеры, мониторы, ревизоры, фильтры) и назначению (базовая защита, комплексные пакеты, облачные решения).

3.2 Сигнатура файла – это уникальная последовательность байтов в начале файла, используемая для идентификации его типа и формата, вне зависимости от расширения. Она позволяет ОС и программам безошибочно определять тип содержимого.

4 Вывод

4.1 В ходе лабораторной работы изучены методы антивирусной защиты ПО.

Лабораторная работа №11
Изучение методов авторизации и аутентификации в настольных приложениях

1 Цель работы

1.1 Познакомиться с методами авторизации и аутентификации в настольных приложениях.

2 Ход работы

2.1 Разработали оконное приложение, использующее аутентификацию пользователя Windows;

2.2 Разработали оконное приложение, использующее авторизацию пользователя через БД с распределением прав на три роли.

3 Ответы на контрольные вопросы

3.1 Аутентификация – это процесс удостоверения личности пользователя или системы, чтобы подтвердить их право на доступ к защищенным ресурсам.

3.2 Авторизация – это процесс определения прав пользователя или системы после успешной аутентификации. Он регулирует, к каким ресурсам и операциям пользователь имеет доступ согласно своей роли или правам.

4 Вывод

4.1 В ходе лабораторной работы изучены методы авторизации и аутентификации в настольных приложениях.

Лабораторная работа №12
Изучение методов авторизации и аутентификации в веб-приложениях

1 Цель работы

1.1 Познакомиться с методами авторизации и аутентификации в веб-приложениях.

2 Ход работы

2.1 Разработали веб-приложение, использующее авторизацию пользователя при помощи JWT с распределением прав на две роли;

2.2 Разработали веб-приложение, использующее авторизацию пользователя при помощи стороннего сервиса.

3 Ответы на контрольные вопросы

3.1 JWT (JSON Web Token) – это открытый стандарт для безопасной передачи данных между сторонами в виде компактного JSON-объекта.

3.2 OAuth – это открытый протокол авторизации, позволяющий приложениям получать ограниченный доступ к данным пользователя на других сервисам без передачи логина и пароля.

4 Вывод

4.1 В ходе лабораторной работы изучены методы авторизации и аутентификации в веб-приложениях.

Лабораторная работа №13
Применение алгоритмов хэширования данных

1 Цель работы

1.1 Познакомиться с методами применения алгоритмов хэширования данных.

2 Ход работы

2.1 Разработали оконное приложение для вычисления хэш-сумм файлов;

2.2 Модифицировали приложение для применения криптографической соли при хэшировании файлов.

3 Ответы на контрольные вопросы

3.1 Популярные алгоритмы хэширования включают SHA-256 (блокчейн, SSL), MD5 (устаревший, для проверок), SHA-3 (современный стандарт), bcrypt и Argon2 (для паролей).

3.2 Криптографическая соль – это случайное значение, добавляемое к исходному паролю перед его хэшированием. Она используется для предотвращения атак с помощью таблиц радуги, так как с помощью соли один и тот же пароль будет иметь разные хэш-значения. Это повышает безопасность хранения паролей, усложняя перебор и кражу данных.

4 Вывод

4.1 В ходе лабораторной работы изучены методы применения алгоритмов хэширования данных.

Лабораторная работа №14
Применение алгоритмов шифрования данных

1 Цель работы

1.1 Познакомиться с методами применения алгоритмов шифрования данных.

2 Ход работы

2.1 Разработали оконное приложение-мессенджер для обмена зашифрованными сообщениями со вторым экземпляром приложения;

2.2 Модифицировали приложение таким образом, чтобы клиентские приложения обменивались зашифрованными сообщениями с использованием ключа шифрования, который должен шифроваться ассиметричным алгоритмом шифрования.

3 Ответы на контрольные вопросы

3.1 Преимущества симметричных алгоритмов шифрования:

- высокая скорость обработки данных, что подходит для шифрования больших объемов информации;
- простота реализации;
- хорошо подходит для защиты данных в закрытых системах и для шифрования файлов.

Недостатки:

- необходимость безопасно обмениваться секретным ключом, что усложняет коммуникацию на большие расстояния и в открытых сетях;
- не подходит для обмена информацией между неконтролируемыми сторонами без защищенного канала.

3.2 Преимущества ассиметричных алгоритмов шифрования:

- позволяют безопасно обмениваться ключами;
- публичный ключ может быть опубликован, а приватный – храниться в тайне;
- обеспечивают возможность цифровой подписи и аутентификации;
- улучшена безопасность при работе с открытыми сетями и обмене данными.

Недостатки:

- значительно медленнее симметричных алгоритмов, что делает их менее подходящими для шифрования больших объемов данных;
- более сложная и ресурсоемкая реализация;
- требуют управления парой ключей и защищенной инфраструктуры для их хранения.

4 Вывод

4.1 В ходе лабораторной работы изучены методы применения алгоритмов шифрования данных.

Лабораторная работа №15
Реализация защиты ПО от нелегального копирования

1 Цель работы

1.1 Познакомиться с методами защиты ПО от нелегального копирования.

2 Ход работы

2.1 Разработали приложение, требующее активацию с помощью лицензионного ключа при первом запуске;

2.2 Модифицировали приложение таким образом, чтобы активация приложения производилась с привязкой к оборудованию пользователя.

3 Ответы на контрольные вопросы

3.1 Методы защиты ПО от нелегального копирования применяются для охраны авторских прав разработчиков и правообладателей.

3.2 Методы защиты ПО от нелегального копирования

- встроенные механизмы проверки лицензии (серийные номера или ключи активации);
- аппаратные ключи или dongle – физические устройства, которые должны быть подключены для работы программы;
- использование проверок подлинности через онлайн-сервисы или серверы лицензий;
- применение криптографических методов для проверки целостности и авторизации;
- защита файлами и шифрование, чтобы усложнить модификацию и реверс-инжиниринг.

4 Вывод

4.1 В ходе лабораторной работы изучены методы защиты ПО от нелегального копирования.