

| |
|--|
| <p>(встроенные в среду программирования Visual Studio, D-Unit, N-Unit, прочие). Интеграционное тестирование (Integration Testing) .</p> <p>Тестирование объектно-ориентированных систем.</p> <p>Тестирование классов объектов. Интеграция объектов. Особенности тестирования классов (тестирование наследования и проч.). Варианты использования, диаграммы Use Case. Применение Use Cases в процессе тестирования. Создание тестового набора на основе вариантов использования. Пример тестирования класса.</p> <p>Системное Тестирование (System Testing). Задачи и цели системного тестирования. Виды системного тестирования.</p> <p>Приемочное Тестирование (Acceptance Testing). Прочее.</p> <p>Лекция 2 (2 часа). Тема 2. Специальные виды тестирования программного обеспечения. Нагрузочное тестирование.</p> <p>Терминология. Основные виды тестирования производительности. Основные принципы нагрузочного тестирования. Цели нагрузочного тестирования. Этапы проведения нагрузочного тестирования. Инструменты нагрузочного тестирования (Apache Jmeter и прочие). Стресс-тестирование.</p> <p>Тестирование пользовательского интерфейса</p> <p>Задачи и цели тестирования пользовательского интерфейса . Функциональное тестирование пользовательских интерфейсов.</p> <p>Проверка требований к пользовательскому интерфейсу. Методы проведения тестирования пользовательского интерфейса, повторяемость тестирования пользовательского интерфейса. Тестирование удобства использования пользовательских интерфейсов.</p> <p>Тестирование пользовательского интерфейса на основе состояний и диаграммы переходов состояний. Методика тестирования пользовательского интерфейса.</p> <p>Тестирование Веб-приложений и мобильных приложений.</p> <p>Особенности веб-приложений как объекта тестирования. Подходы к функциональному тестированию Веб-приложений.</p> <p>Нагрузочное и стресс-тестирование веб-приложений. Инструменты тестирования веб-приложений. Тестирование мобильных приложений</p> <p>Тестирование баз данных. Тестирование модели, целостности, связности, непротиворечивости, тестирование атрибутов, сценариев, проч. Функциональное тестирование. Нагрузочное и стресс-тестирование баз данных.</p> <p>Тестирование надежности программного обеспечения и методы повышения надежности.</p> <p>Основные показатели надежности программного обеспечения.</p> <p>Математические модели оценки надежности ПО. Основные показатели надежности программного обеспечения. Основные понятия в проблематике надежности ПС. Классификация моделей надежности. Марковские и пуассоновские модели надежности.</p> <p>Модель Джелинского-Моранды. Модель Шика-Уолвертона.</p> <p>Тестирование качества и управление качеством программного обеспечения.</p> <p>Качество программного обеспечения (Software Quality). Требования к качеству программного обеспечения. Модели и</p> |
|--|

| | |
|---|---|
| | характеристики качества. Метрики, способы оценки качества. Процессы управления качеством программного обеспечения. |
| 2 | <p>лабораторные работы 4 шт. по 2 часа, дополнительное аудиторное время на защиту – 2 часа:</p> <p>Лабораторная работа № 1 (2 часа). Структурное тестирование программного обеспечения. Функциональное тестирование программного обеспечения. Цель работы: Изучение методов тестирования программного обеспечения. Задачи: Выполнение ручного тестирования программ. Выполнение структурного тестирования программ следующими методами: тестирование базового пути; тестирование условий; тестирование циклов; тестирование потоков данных. Выполнение функционального тестирования программ следующими методами: разбиение на классы эквивалентности; анализ граничных значений; анализ причинно-следственных связей; предположение об ошибке.</p> <p>Изучение инструментов автоматизированного модульного (Unit) тестирования. Цель: получение начальных навыков применения инструментария автоматизированного тестирования программного кода методом белого ящика. Задачи: Изучить средства автоматизированного тестирования, встроенные в MS Visual Studio, отдельные кроссплатформенные инструменты. Подготовить план тестирования методами белого ящика (по вариантам) и создать тестовый проект для заданной программы. Получить и проанализировать результаты тестирования.</p> <p>Лабораторная работа №2 (2 часа). Тестирование объектно-ориентированного программного обеспечения. Цель: изучение методов тестирования объектно-ориентированного программного обеспечения, получение навыков применения инструментов автоматизированного тестирования средствами MS Visual Studio (программирование тестового проекта). Задачи: Разработать объектно-ориентированное приложение. Предусмотреть создание интерфейсов и не менее трех уровней иерархии классов. Выполнить объектно-ориентированное тестирование разработанного приложения: тестирование методов классов (Unit – тестирование методом белого ящика), тестирование наследования, составление тест-кейсов для функционального тестирования методом черного ящика на основе диаграмм UseCases.</p> <p>Разработка и тестирование web-сайта. Функциональное тестирование и отладка разработанного приложения с использованием стратегий тестирования интернет приложений. Цель: получить навык функционального тестирования веб-приложений. Задачи: согласовать техническое задание на разработку сайта, выполнить реализацию сайта, установить на локальный веб-сервер APACHE. Составить план тестирования, выполнить функциональное тестирование методами черного ящика.</p> <p>Лабораторная работа №3 (2 часа). Тестирование производительности web – сайтов. Цель: получить навык тестирования производительности веб-сайтов с помощью инструментов автоматизированного тестирования (APACHE JMETER, онлайн-сервисы). Задачи: Нагрузочное тестирование (load testing), сбор показателей и определение производительности и времени отклика веб-сайта в ответ на генерируемые внешние запросы (нагрузку); исследование времени отклика системы на высоких или пиковых нагрузках</p> |

| | |
|---|---|
| | <p>(стресс-тестирование).</p> <p>Разработка и тестирование базы данных.</p> <p>Цель: получить навык тестирования баз данных. Задачи: Согласовать техническое задание на разработку базы данных, реализовать БД в СУБД mysql. Выполнить тестирование БД на основе методов, изложенных в лекции, за исключением нагрузочного тестирования.</p> <p>Лабораторная работа №4 (2 часа). Тестирование производительности базы данных с помощью APACHE JMETER. Цель: получить навык тестирования производительности баз данных с помощью инструментов автоматизированного тестирования. Задачи: сделать замеры времени выполнения операций при определенных, при максимальных, минимальных и средних нагрузках; определить количество пользователей, которые одновременно могут работать с продуктом; определить границы приемлемой производительности при увеличении нагрузки; исследовать производительность при больших, максимальных и стрессовых нагрузках.</p> <p>Тестирование документации.</p> <p>Цель: получить навык разработки и тестирования документации в процессах тестирования программного обеспечения. Задачи: подготовить пакет документации для разработанных ранее веб-сайта и базы данных, включая: технические задания, технические описания, программу и методику испытаний, протокол испытаний и отчет об испытаниях. Разработать программу (план) тестирования документации на соответствие комплектности, содержанию и прочим критериям, рассмотренным в лекции. Дополнить программу и методику испытаний соответствующим разделом; результаты работы внести в отчет и протокол.</p> <p>Дополнительное аудиторное время для защит лабораторных работ – 2 часа.</p> |
| 3 | |
| 4 | <p>Самостоятельная работа студентов:</p> <ol style="list-style-type: none"> 1. Самостоятельное изучение материала: <ul style="list-style-type: none"> • К теме 1: Подробное рассмотрение классификаций видов тестирования: Классификация по запуску кода на исполнение. Классификация по доступу к коду и архитектуре приложения. Классификация по степени автоматизации. Классификация по уровню детализации приложения. Классификация по принципам работы с приложением. Классификация по природе приложения. Классификация по фокусировке на уровне архитектуры приложения. Классификация по привлечению конечных пользователей. Классификация по степени формализации . Классификация по целям и задачам. Классификация по техникам и подходам. Классификация по моменту выполнения (хронологии). • К теме 1: Регрессионное тестирование программного обеспечения. <p>Цели и задачи регрессионного тестирования. Виды регрессионного тестирования. Управляемое регрессионное тестирование.</p> |

| |
|--|
| <p>Обоснование корректности метода отбора тестов. Классификация тестов при отборе. Возможности повторного использования тестов. Классификация выборочных методов.</p> <ul style="list-style-type: none">• К теме 1: Основные понятия отладки программного обеспечения. Принципы отладки ПО. Методы отладки программного обеспечения. Виды отладки программного обеспечения. Автономная отладка программного обеспечения. Комплексная отладка программного обеспечения. Методы отладки параллельных программ. Анализ инструментов отладки параллельных программ.• К теме 3: Модели тестирования надежности ПО, основанные на методе "посева" и разметки ошибок, и модели на основе учета структуры входных данных. Модель Нельсона. Применение последовательного анализа Вальда для снижения количества прогонов программы.• К теме 3: Методы повышения надежности программ за счет резервирования и оценка эффективности их применения. Влияние избыточности на повышение надежности программ. Эффективность применения избыточности для повышения надежности комплексов программ. Влияние оперативного контроля и восстановления на производительность. Методы программного восстановления. Методы обеспечения надежности комплексов программ при сопровождении. <ol style="list-style-type: none">2. Подготовка к выполнению и защите лабораторных работ (4 работы).3. Подготовка к экзамену по дисциплине с использованием оценочных материалов, приведенных в разделе 6 настоящей РПД |
|--|

Текущий контроль:

- проверка дополнительных теоретических материалов;
- проверка отчетов по лабораторным работам и защита лабораторных работ.

Результаты текущего контроля фиксируются с использованием трехбалльной системы (0, 1, 2) при проведении контрольных недель по графику филиала в течение семестра, а также учитываются преподавателем при осуществлении промежуточной аттестации по настоящей дисциплине.