



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Магнитогорский государственный технический университет им. Г.И. Носова»



УТВЕРЖДАЮ
Директор ИЭиАС
С.И. Лукьянов

26.02.2020 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)

***ТЕХНОЛОГИИ ТЕСТИРОВАНИЯ ПРОГРАММНЫХ ПРОДУКТОВ,
ИСПОЛЬЗУЕМЫЕ КОМПАНИЕЙ-РАЗРАБОТЧИКОМ***

Направление подготовки (специальность)

09.04.01 Информатика и вычислительная техника

Направленность (профиль/специализация) программы

Программное обеспечение средств вычислительной техники и автоматизированных систем

Уровень высшего образования - магистратура

Форма обучения
очная

Институт/ факультет	Институт энергетики и автоматизированных систем
Кафедра	Вычислительной техники и программирования
Курс	2
Семестр	4

Магнитогорск
2020 год

Рабочая программа составлена на основе ФГОС ВО - магистратура по направлению подготовки 09.04.01 Информатика и вычислительная техника (приказ Минобрнауки России от 19.09.2017 г. № 918)

Рабочая программа рассмотрена и одобрена на заседании кафедры вычислительной техники и программирования

19.02.2020 г. протокол № 5

Зав. кафедрой  О.С. Логунова

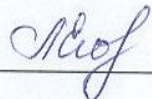
Рабочая программа одобрена методической комиссией ИЭ и АС

26.02.2020 г. протокол № 5

Председатель  С.И. Лукьянов

Рабочая программа составлена:


доцент кафедры ВТ и П, канд. техн. наук

 Л.Г.Егорова

Рецензент:

Начальник отдела технологических платформ

ООО Компас Плюс, канд. техн. наук

 Д.С.Сафонов

Лист актуализации рабочей программы

Рабочая программа пересмотрена, обсуждена и одобрена для реализации в 2021 - 2022 учебном году на заседании кафедры вычислительной техники и программирования

Протокол от ____ 20__ г. № ____
Зав. кафедрой _____ О.С. Логунова

Рабочая программа пересмотрена, обсуждена и одобрена для реализации в 2022 - 2023 учебном году на заседании кафедры вычислительной техники и программирования

Протокол от ____ 20__ г. № ____
Зав. кафедрой _____ О.С. Логунова

1 Цели освоения дисциплины (модуля)

Цель изучения курса «Технологии тестирования программных продуктов, используемых компанией разработчиком» - ознакомление студентов с основными видами и методами тестирования программного обеспечения. В курсе изучаются способы обеспечения качества программного продукта, классы критериев тестирования, разновидности тестирования, модульное, интеграционное и системное тестирование, общие принципы автоматизации тестирования, издержки тестирования.

2 Место дисциплины (модуля) в структуре образовательной программы

Дисциплина Технологии тестирования программных продуктов, используемые компанией-разработчиком входит в обязательную часть учебного плана образовательной программы.

Для изучения дисциплины необходимы знания (умения, владения), сформированные в результате изучения дисциплин/ практик:

Информационно-управляющие системы

Конфигурирование на платформе TranzAxis

Автоматизированная система TranzWare для розничных банковских процессов

Oracle Database: продвинутые аспекты программирования и настройки производительности

Проектирование и тестирование сложных пользовательских интерфейсов

Современные розничные финансовые платформы на примере TranzAxis

Программное обеспечение для управления проектами в финансовой индустрии

Стандарты PA/PCI DSS в финансовой индустрии

Технологии PL/SQL

Знания (умения, владения), полученные при изучении данной дисциплины будут необходимы для изучения дисциплин/практик:

Автоматизированная система TranzWare для розничных банковских процессов

Выполнение и защита выпускной квалификационной работы

Конфигурирование на платформе TranzAxis

Производственная - технологическая (проектно-технологическая) практика

Производственная-преддипломная практика

3 Компетенции обучающегося, формируемые в результате освоения дисциплины (модуля) и планируемые результаты обучения

В результате освоения дисциплины (модуля) «Технологии тестирования программных продуктов, используемые компанией-разработчиком» обучающийся должен обладать следующими компетенциями:

Код индикатора	Индикатор достижения компетенции
ОПК-8	Способен осуществлять эффективное управление разработкой программных средств и проектов.
ОПК-8.1	Оценивает эффективность управления разработкой программных средств и проектов

4. Структура, объём и содержание дисциплины (модуля)

Общая трудоемкость дисциплины составляет 4 зачетных единиц 144 академических часов, в том числе:

- контактная работа – 52,8 академических часов;
- аудиторная – 50 академических часов;
- внеаудиторная – 2,8 академических часов
- самостоятельная работа – 55,5 академических часов;
- подготовка к экзамену – 35,7 академических часов

Форма аттестации - экзамен

Раздел/ тема дисциплины	Семестр	Аудиторная контактная работа (в академических часах)			Самостоятельная работа студента	Вид самостоятельной работы	Форма текущего контроля успеваемости и промежуточной аттестации	Код компетенции
		Лек.	лаб. зан.	практ. зан.				
1. Основные понятия тестирования программных продуктов								
1.1 Критерии выбора тестов. Оценка оттестированности проекта: метрики и методика оценки. Графовые модели проекта, Метрики оценки оттестированности проекта. Примеры плоской и иерархической моделей проекта.		2	6/3И		5,5	1. Подготовка к лабораторным занятиям 2. Выполнение лабораторных работ 3. Самостоятельное изучение учебной и научной литературы	1. Беседа - обсуждение 2. Проверка индивидуальных заданий 3. Устный опрос.	ОПК-8.1
1.2 Типы тестов и их роль в процессе разработки программного обеспечения. Требования к идеальному критерию тестирования и классы частных критериев. Особенности применения структурных и функциональных критериев. Особенности применения методов стохастического тестирования и метод оценки скорости выявления ошибок. Мутационный критерий и техника работы с ним.	4	2	6/3И		10	1. Подготовка к лабораторным занятиям 2. Выполнение лабораторных работ 3. Самостоятельное изучение учебной и научной литературы	1. Беседа - обсуждение 2. Проверка индивидуальных заданий 3. Устный опрос.	ОПК-8.1
Итого по разделу		4	12/6И		15,5			

2. Виды тестирования программных продуктов								
2.1 Особенности модульного тестирования. Подходы к тестированию на основе потока управления, потока данных. Динамические и статические методы при структурном подходе. Пример модульного тестирования. Взаимосвязь сборки модулей и методов интеграционного тестирования. Подходы монолитного, инкрементального, нисходящего и восходящего тестирования.	4	2	6/3И		10	1. Подготовка к лабораторным занятиям 2. Выполнение лабораторных работ 3. Самостоятельное изучение учебной и научной литературы	1. Беседа - обсуждение 2. Проверка индивидуальных заданий 3. Устный опрос.	ОПК-8.1
2.2 Интеграционное тестирование и его особенности для объектно-ориентированного программирования. Модель объектно-ориентированной программы, использующая понятие Р-путей и ММ-путей. Оценки сложности тестирования и методика тестирования объектно-ориентированной программы. Особенности интеграционного тестирования в процедурном программировании. Пример интеграционного тестирования.	4	2	6/3И		10	1. Подготовка к лабораторным занятиям 2. Выполнение лабораторных работ 3. Самостоятельное изучение учебной и научной литературы	1. Беседа - обсуждение 2. Проверка индивидуальных заданий 3. Устный опрос.	ОПК-8.1
2.3 Системное и регрессионное тестирование. Задачи и категории тестов, применяемые в системном тестировании. Пример системного тестирования. Регрессионное тестирование и комбинирование различных уровней тестирования.		2	6/3И		10	1. Подготовка к лабораторным занятиям 2. Выполнение лабораторных работ 3. Самостоятельное изучение учебной и научной литературы	1. Беседа - обсуждение 2. Проверка индивидуальных заданий 3. Устный опрос.	ОПК-8.1
Итого по разделу	6	18/9И		30				
3. Автоматизация тестирования программных продуктов								

3.1 Структура тестового набора для автоматического прогона. Структура инструментальной системы автоматизации тестирования. Издержки и эффективность различных методов тестирования.	4	10/5И	10	1. Подготовка к лабораторным занятиям 2. Выполнение лабораторных работ 3. Самостоятельное изучение учебной и научной литературы	1. Беседа - обсуждение 2. Проверка индивидуальных заданий 3. Устный опрос.	ОПК-8.1
Итого по разделу		10/5И	10			
Итого за семестр	10	40/20И	55,5		экзамен	
Итого по дисциплине	10	40/20И	55,5		экзамен	

5 Образовательные технологии

1. Традиционные образовательные технологии, ориентированные на организацию образовательного процесса и предполагающие прямую трансляцию знаний от преподавателя к студенту.

Формы учебных занятий с использованием традиционных технологий:

Лабораторная работа – организация учебной работы с реальными материальными и информационными объектами, экспериментальная работа с аналоговыми моделями реальных объектов.

2. Технологии проблемного обучения – организация образовательного процесса, которая предполагает постановку проблемных вопросов, создание учебных проблемных ситуаций для стимулирования активной познавательной деятельности студентов.

Формы учебных занятий с использованием технологий проблемного обучения:

Практическое занятие в форме практикума – организация учебной работы, направленная на решение комплексной учебно-познавательной задачи, требующей от студента применения как научно-теоретических знаний, так и практических навыков.

6 Учебно-методическое обеспечение самостоятельной работы обучающихся

Представлено в приложении 1.

7 Оценочные средства для проведения промежуточной аттестации

Представлены в приложении 2.

8 Учебно-методическое и информационное обеспечение дисциплины (модуля)

а) Основная литература:

1. Трояновский, В. М. Программная инженерия информационно-управляющих систем в свете прикладной теории случайных процессов : учеб. пособие / В.М. Трояновский. — Москва : ИНФРА-М, 2019. — 325 с. + Доп. материалы [Электронный ресурс; Режим доступа: <http://new.znanium.com>]. —(Высшее образование: Магистратура). — www.dx.doi.org/10.12737/textbook_5ad88bf5c35cd8.81685342. - ISBN 978-5-8199-0824-2. - Текст : электронный. - URL:

<https://znanium.com/catalog/product/1003316>

(дата обращения: 28.10.2020). – Режим доступа: по подписке.

2. Плаксин, М. А. Тестирование и отладка программ для профессионалов будущих и настоящих / М. А. Плаксин. — 4-е изд., электрон. — Москва : Лаборатория знаний, 2020. — 170 с. - ISBN 978-5-00101-810-0. - Текст : электронный. - URL:

<https://znanium.com/catalog/product/1093870>

б) Дополнительная литература:

1. Антамошкин, О. А. Программная инженерия. Теория и практика [Электронный ресурс] : учебник / О. А. Антамошкин. - Красноярск: Сиб. Федер. ун-т, 2012. - 247 с. - ISBN 978-5-7638-2511-4. - Текст : электронный. - URL:

<https://znanium.com/catalog/product/492527>

(дата обращения: 28.10.2020). – Режим доступа: по подписке.

в) Методические указания:

Представлены в приложении 1.

г) Программное обеспечение и Интернет-ресурсы:

Программное обеспечение

Наименование ПО	№ договора	Срок действия лицензии
MS Windows 7 Professional(для классов)	Д-1227-18 от 08.10.2018	11.10.2021
MS Office 2007 Professional	№ 135 от 17.09.2007	бессрочно
Kaspersky Endpoint Security для бизнеса-Стандартный	Д-300-18 от 21.03.2018	28.01.2020
MS Visual Studio 2010 Professional(для класса)	Д-1227-18 от 08.10.2018	11.10.2021
SCO OpenServer	свободно распространяемое ПО	бессрочно
Oracle Open JDK	свободно распространяемое ПО	бессрочно
MS Visual Studio 2017 Community Edition	свободно распространяемое ПО	бессрочно
MS Visual Studio Code	свободно распространяемое ПО	бессрочно
MS Visual Studio 2013 Professional(для класса)	Д-1227-18 от 08.10.2018	11.10.2021

Профессиональные базы данных и информационные справочные системы

Название курса	Ссылка
Национальная информационно-аналитическая система – Российский индекс научного цитирования (РИНЦ)	URL: https://elibrary.ru/project_risc.asp
Информационная система - Единое окно доступа к информационным ресурсам	URL: http://window.edu.ru/

Федеральное государственное бюджетное учреждение «Федеральный институт промышленной собственности»	URL: http://www1.fips.ru/
--	--

9 Материально-техническое обеспечение дисциплины (модуля)

Материально-техническое обеспечение дисциплины включает:

1. Лекционная аудитория ауд. 282. Мультимедийные средства хранения, передачи и представления информации.
2. Компьютерные классы Центра информационных технологий ФГБОУ ВО «МГТУ». Персональные компьютеры, объединенные в локальные сети с выходом в Internet, оснащенные современными программно-методическими комплексами для решения задач в области информатики и вычислительной техники.
3. Аудитории для самостоятельной работы: компьютерные классы; читальные залы библиотеки. Все классы УИТ и АСУ с персональными компьютерами, выходом в Интернет и с доступом в электронную информационно-образовательную среду университета.
4. Аудиторий для групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации. Ауд. 282 и классы УИТ и АСУ.
5. Помещения для самостоятельной работы обучающихся, оснащенных компьютерной техникой с возможностью подключения к сети «Интернет» и наличием доступа в электронную информационно-образовательную среду организации. Классы УИТ и АСУ.
6. Помещения для хранения и профилактического обслуживания учебного оборудования. Центр информационных технологий – ауд. 372.

Приложение 1

Учебно-методическое обеспечение самостоятельной работы обучающихся

Лабораторная работа 1

Разработка спецификаций системных требований программному продукту

к

Цель работы: изучение требований к создаваемому программному продукту, разработка технического задания

Задание

1 Изучить нормативные документы по разработке технического задания на разработку программного продукта.

2 Разработать техническое задание на программный продукт по заданному варианту.

Техническое задание представляет собой документ, в котором сформулированы основные цели разработки, требования к программному продукту, определены сроки и этапы разработки и регламентирован процесс приемно-сдаточных испытаний. В разработке технического задания участвуют как представители заказчика, так и представители исполнителя. В основе этого документа лежат исходные требования заказчика, анализ передовых достижений техники, результаты выполнения научно-исследовательских работ, предпроектных исследований, научного прогнозирования и т. п.

Основные факторы, определяющие характеристики разрабатываемого программного обеспечения:

- исходные данные и требуемые результаты, которые определяют функции программы или системы;
- среда функционирования (программная и аппаратная); может быть задана, а может выбираться для обеспечения параметров, указанных в техническом задании;
- возможное взаимодействие с другим программным обеспечением или специальными техническими средствами - также может быть определено, а может выбираться исходя из набора выполняемых функций.

Разработка технического задания выполняется в следующей последовательности. Прежде всего, устанавливается набор выполняемых функций, а также перечень и характеристики исходных данных. Затем определяют перечень результатов, их характеристики и способы представления.

Далее уточняют среду функционирования программного обеспечения: конкретную комплектацию и параметры технических средств, версию используемой операционной системы и, возможно, версии и параметры другого установленного программного обеспечения, с которым предстоит взаимодействовать будущему программному продукту.

В случаях, когда разрабатываемое программное обеспечение собирает и хранит некоторую информацию или включается в управление каким-либо техническим процессом, необходимо также четко регламентировать действия программы в случае сбоев оборудования и энергоснабжения.

На техническое задание существует стандарт ГОСТ 19.201-78.

«Техническое задание. Требования к содержанию и оформлению». В соответствии с этим стандартом техническое задание должно содержать следующие разделы:

- введение;
- основания для разработки;
- назначение разработки;
- требования к программе или программному изделию;

- требования к программной документации;
- технико-экономические показатели;
- стадии и этапы разработки;
- порядок контроля и приемки.

При необходимости допускается в техническое задание включать приложения.

Контрольные вопросы

1 Что понимают под технологичностью программного обеспечения? Почему?

2 Какие типы программных продуктов можно выделить? Чем они различаются?

3 Назовите основные эксплуатационные требования к программным продуктам. Какими средствами и приемами обеспечивается каждый из них? Для каких типов программных систем целесообразно указывать каждый из них?

4 В каких ситуациях необходимы предпроектные исследования? Какие вопросы при этом решают? Что получают в результате таких исследований?

5 Назовите, какой раздел технического задания можно считать основным и почему? Какую информацию должны содержать остальные разделы? В чем основная сложность разработки технического задания?

Варианты заданий

1. Платформа для удаленной идентификации.
2. Платформа быстрых платежей
3. Платформа-маркетплейс для финансовых услуг и продуктов
4. Платформа для регистрации финансовых сделок
5. Система передачи финансовых сообщений
6. Создание сквозного идентификатора клиента
7. Создание платформы для облачных сервисов
8. Создание платформы на основе технологии распределенных реестров
9. Платформа для электронного взаимодействия.

Лабораторная работа 2

Виды тестирования. Планирование тестирования

Цель работы: изучить классификацию видов тестирования, практически закрепить эти знания путем генерации тестов различных видов, научиться планировать тестовые активности в зависимости от специфики поставляемой на тестирование функциональности.

Тестирование – процесс, направленный на оценку корректности, полноты и качества разработанного программного обеспечения.

Тестирование можно классифицировать по очень большому количеству признаков. Далее приведен обобщенный список видов тестирования по различным основаниям.

Задание

1. Выполнить генерацию тестов различных видов для конкретного варианта.
3. Спланировать тестовые активности для следующих задач:
 - 3.1 Поставлен на тестирование модуль 1, модуль 2, модуль 3.
 - 3.2 Проведены исправления (fix) для заведенных дефектов, доставлена новая функциональность – модуль 4.
 - 3.3 Заказчик решил расширять рынки сбыта и просит осуществить поддержку для Великобритании (кроме уже существующей в Малазии).
 - 3.4 Заказчик хочет убедиться, что ПО держит нагрузку в 2000 пользователей.
4. Оформить отчет и защитить лабораторную работу.

Контрольные вопросы

Что такое тестирование?

Какие существуют типы тестов по покрытию? Дайте характеристику каждому.

Какие существуют тестовые активности? Дайте характеристику каждому.

Какие существуют типы тестов знанию кода? Дайте характеристику каждому.

Какие существуют типы тестов по степени автоматизации? Дайте характеристику каждому.

Какие существуют типы тестов по изолированности компонентов? Дайте характеристику каждому.

Какие существуют типы тестов по подготовленности? Дайте характеристику каждому.

Какие существуют типы тестов по месту и времени проведения? Дайте характеристику каждому.

Какие существуют типы тестов по объекту тестирования? Дайте характеристику каждому.

Какие существуют типы функциональных тестов? Дайте характеристику каждому.

Какие существуют типы нефункциональных тестов? Дайте характеристику каждому.

Какие этапы составляют процесс тестирования?

Что происходит на этапе изучения и анализа предмета тестирования?

Что происходит на этапе планирования тестирования?

Что происходит на этапе исполнения тестирования?

Какие типы тестов выполняют для первой поставки программного продукта?

Какие типы тестов выполняют для последующих поставок программного продукта?

Лабораторная работа 3

Технологии разработки программного обеспечения: Разработка через тестирование"

Цель работы

Знакомство с технологией "разработка через тестирование". Изучение инструментов позволяющих применять данную технологию.

Общие сведения

Разработка через тестирование (англ. *test-driven development, TDD*) — техника разработки программного обеспечения, которая основывается на повторении очень коротких циклов разработки: сначала пишется тест, покрывающий желаемое изменение, затем пишется код, который позволит пройти тест, и под конец проводится рефакторинг нового кода к соответствующим стандартам.

TDD требует от разработчика создания автоматизированных модульных тестов, определяющих требования к коду непосредственно перед написанием самого кода. Тест содержит проверки условий, которые могут либо выполняться, либо нет. Когда они выполняются, говорят, что тест пройден. Прохождение теста подтверждает поведение, предполагаемое программистом. Разработчики часто пользуются библиотеками для тестирования (англ. *testing frameworks*) для создания и автоматизации запуска наборов тестов. На практике модульные тесты покрывают критические и нетривиальные участки кода. Это может быть код, который подвержен частым изменениям, код, от работы которого зависит работоспособность большого количества другого кода, или код с большим количеством зависимостей.

Среда разработки должна быстро реагировать на небольшие модификации кода. Архитектура программы должна базироваться на использовании множества сильно связанных компонентов, которые слабо сцеплены друг с другом, благодаря чему

тестирование кода упрощается. TDD не только предполагает проверку корректности, но и влияет на дизайн программы. Опираясь на тесты, разработчики могут быстрее представить, какая функциональность необходима пользователю. Таким образом, детали интерфейса появляются задолго до окончательной реализации решения.

Задание

1. Изучить библиотеки для тестирования
2. Рассмотреть применение NUnit, ReSharper.

Контрольные вопросы

- Что такое динамическое тестирование?
- Что такое GUI-тестирование (GUI Testing)?
- Что такое формальное тестирование?
- Что такое тестирование на основе рисков?
- Что такое раннее тестирование?
- Что такое исчерпывающее тестирование?
- Что такое скопление дефектов?
- Что такое статическое тестирование?
- Что такое позитивное тестирование?
- Что такое негативное тестирование?
- Что такое сквозное тестирование (end-to-end)?

Лабораторная работа 4 ***Модульное тестирование***

Цель работы

Овладение навыками модульного тестирования.

Общие сведения

Модульное тестирование - это тестирование программы на уровне отдельно взятых модулей, функций или классов. Цель модульного тестирования состоит в выявлении локализованных в модуле ошибок в реализации алгоритмов, а также в определении степени готовности системы к переходу на следующий уровень разработки и тестирования. Модульное тестирование проводится по принципу "белого ящика", то есть основывается на знании внутренней структуры программы, и часто включает те или иные методы анализа покрытия кода.

Модульное тестирование обычно подразумевает создание вокруг каждого модуля определенной среды, включающей заглушки для всех интерфейсов тестируемого модуля. Некоторые из них могут использоваться для подачи входных значений, другие для анализа результатов, присутствие третьих может быть продиктовано требованиями, накладываемыми компилятором и сборщиком.

На уровне модульного тестирования проще всего обнаружить дефекты, связанные с алгоритмическими ошибками и ошибками кодирования алгоритмов, типа работы с условиями и счетчиками циклов, а также с использованием локальных переменных и ресурсов. Ошибки, связанные с неверной трактовкой данных, некорректной реализацией интерфейсов, совместимостью, производительностью и т.п. обычно пропускаются на уровне модульного тестирования и выявляются на более поздних стадиях тестирования.

Задание

1. Выбор и согласование объекта тестирования
2. Разработка плана тестирования.
3. Тестирование (инспекция) проектной документации и кода.

4. Реализация модульных тестов, запуск.
5. Анализ результатов тестирования и подготовка отчета.

Контрольные вопросы

- Что такое модульное тестирование?
- Что такое тестирование на уровне системы?
- Что такое тестирование на основе потока управления?
- Что такое тестирование на основе потока данных.
- В чем отличие динамического метода тестирования от статического?
- Какая взаимосвязь сборки модулей и методов интеграционного тестирования?
- Что такое монолитное тестирование
- Что такое инкрементальное тестирование
- Что такое нисходящее тестирование?
- Что такое тестирование восходящее тестирование?

Лабораторная работа 5 ***Интеграционное тестирование***

Цель работы

Овладение навыками интеграционного тестирования.

Общие сведения

Интеграционное тестирование называют еще тестированием архитектуры системы. С одной стороны, это название обусловлено тем, что интеграционные тесты включают в себя проверки всех возможных видов взаимодействий между программными модулями и элементами, которые определяются в архитектуре системы – таким образом, интеграционные тесты проверяют полноту взаимодействий в тестируемой реализации системы. С другой стороны, результаты выполнения интеграционных тестов - один из основных источников информации для процесса улучшения и уточнения архитектуры системы, межмодульных и межкомпонентных интерфейсов. Т.е., с этой точки зрения, интеграционные тесты проверяют корректность взаимодействия компонент системы.

В результате проведения интеграционного тестирования и устранения всех выявленных дефектов получается согласованная и целостная архитектура программной системы, т.е. можно считать, что интеграционное тестирование - это тестирование архитектуры и низкоуровневых функциональных требований. Интеграционное тестирование, как правило, представляет собой итеративный процесс, при котором проверяется совокупность модулей, возрастающая от итерации к итерации. В интеграционном тестировании выделяют три метода выполнения: восходящее тестирование; монолитное тестирование; нисходящее тестирование.

Задание

Согласно варианту провести один из методов интеграционного тестирования.

Контрольные вопросы

- В чем суть интеграционного тестирования?
- В чем суть Р-путей и ММ-путей?
- Приведите оценки сложности тестирования объектно-ориентированной программы.
- В чем суть методики тестирования объектно-ориентированной программы?
- В чем особенность интеграционного тестирования в процедурном программировании?
- Что такое тестирование по сценарию использования?

Лабораторная работа 6 **Системное тестирование**

Цель работы

Овладение навыками системного тестирования.

Общие сведения

Системное тестирование - один из самых сложных видов тестирования. На этапе системного тестирования проводится не только функциональное тестирование, но и оценка характеристик качества системы - ее устойчивости, надежности, безопасности и производительности. На этом этапе выявляются многие проблемы внешних интерфейсов системы, связанные с неверным взаимодействием с другими системами, аппаратным обеспечением, неверным распределением памяти, отсутствием корректного освобождения ресурсов и т.п.

После завершения системного тестирования разработка переходит в фазу приемосдаточных испытаний (для программных систем, разрабатываемых на заказ) или в фазу альфа- и бета-тестирования (для программных систем общего применения). Системное тестирование проводится в несколько фаз, на каждой из которых проверяется один из аспектов поведения системы, т.е. проводится один из типов системного тестирования. Все эти фазы могут протекать одновременно или последовательно. Следующий раздел посвящен рассмотрению особенностей каждого из типов системного тестирования на каждой фазе.

Виды системного тестирования:

- 1) функциональное тестирование;
- 2) тестирование производительности;
- 3) нагрузочное или стрессовое тестирование;
- 4) тестирование конфигурации;
- 5) тестирование безопасности;
- 6) тестирование надежности и восстановления после сбоев;
- 7) тестирование удобства использования.

Задание

Согласно варианту провести несколько видов системного тестирования.

Контрольные вопросы

Что такое тестирование на уровне системы?

Что такое регрессионное тестирование?

Перечислите задачи и категории тестов, применяемые в системном тестировании.

Что такое функциональное тестирование?

Что такое тестирование производительности?

Что такое нагрузочное или стрессовое тестирование?

Что такое тестирование конфигурации?

Что такое тестирование безопасности?

Что такое тестирование надежности и восстановления после сбоев?

Что такое тестирование удобства использования?

В чем заключается суть комбинирования различных уровней тестирования?

Лабораторная работа 7

Комплексное тестирование

Задание

1. Выбор и согласование объекта тестирования
2. Разработка плана тестирования.
3. Тестирование (инспекция) проектной документации и кода.
4. Реализация модульных тестов, запуск.
5. Реализация интеграционных тестов, запуск.
6. Реализация системных тестов, запуск.
7. Анализ результатов тестирования и подготовка отчета.

Структура отчета о выполнении тестирования

1. Объект тестирования. Описание объекта тестирования, рамки тестирования, перечень функциональности объекта тестирования. Для каждой функциональности указать ее участие в аттестационном тестировании.
2. Стратегия тестирования.
 - Описание структуры объекта тестирования и связей внутри объекта тестирования (архитектура). Для каждого структурного элемента указать отношение к тестированию.
 - Описание стратегии блочного тестирования (метод проведения, используемое окружение и инструменты, способ оценки результатов).
 - Описание стратегии интеграционного тестирования (схема интеграции, последовательность шагов интеграции с указанием на каждом шаге способа интеграции, метод проведения, используемые окружение и инструменты, способ оценки результатов).
 - Описание стратегии аттестационного тестирования (метод проведения, используемые окружение, инструменты, способ оценки результатов).
 - Описание стратегии выполнения специальных видов тестов (нагрузочное тестирование, тестирование безопасности и т. д.).
 - Условия начала, окончания и перехода между этапами тестирования.
 - Условия возобновления и приостановки выполнения тестов.
3. Детальный план тестов. Перечень блочных, интеграционных, аттестационных и специальных тестов. Для каждого теста необходимо указать:
 - цель теста (описание);
 - тип теста (общий, краевой, негативный, специальный и т.п.);
 - объект тестирования (модуль, интерфейс или функциональность);
 - входные данные;
 - косвенные входные данные, в т. ч. результаты работы функций-заглушек;
 - ожидаемый результат.
4. Пример реализации теста. Метод оценки покрытия тестирования и полученная оценка.
5. Журнал тестирования. Дата, тестировщик, объект тестирования, перечень выполненных тестов с указанием количества запусков, перечень найденных ошибок.
6. Журнал найденных ошибок. Номер отчета об ошибке, дата составления отчета, номер теста, ожидаемый результат, фактический результат.
7. Результаты. Оценка качества исследуемого объекта, оценка результатов тестирования.

Оценочные средства для проведения промежуточной аттестации

а) Планируемые результаты обучения и оценочные средства для проведения промежуточной аттестации:

Код индикатора	Индикатор достижения компетенции	Оценочные средства
ОПК-8 Способен осуществлять эффективное управление разработкой программных средств и проектов		
ОПК-8.1	Оценивает эффективность управления разработкой программных средств и проектов	<p>Перечень теоретических вопросов</p> <ol style="list-style-type: none"> 1. Дайте определение понятию «баг» - в широком смысле и применительно к ПО. 2. Что такое «функциональная спецификация»? 3. Что такое «тестирование ПО»? Какова его цель? 4. Какие существуют основные методики тестирования? В чем их суть? 5. Перечислите основные этапы разработки ПО. Опишите роль и место тестирования в этом процессе. 6. Что такое «тест-кейс» и «тест-комплект»? 7. Какие возможны исходы тест-кейса? 8. Что такое «создание тест-кейса»? 9. Что такое «исполнение тест-кейса»? 10. Структура документально оформленного тест-кейса. Опишите назначение разделов. 11. Перечислите и обоснуйте основные требования к тест-кейсу. 12. Что такое «эффективность тест-комплекта»? 13. Обоснуйте, почему необходимо документально оформлять тест-кейсы? 14. Что такое исчерпывающее тестирование и в каких случаях оно возможно? Приведите примеры. 15. Перечислите основные этапы тестирования. Опишите цель каждого этапа. 16. Структура баг-репорта. Опишите назначение разделов. 17. Что такое багтрекер? Назовите наиболее популярные багтрекеры. 18. Что такое регрессионный баг? 19. Что такое регрессионное тестирование?

Код индикатора	Индикатор достижения компетенции	Оценочные средства
		<p>20. Перечислите основные этапы цикла регрессионного тестирования.</p> <p>21. Обоснуйте принципы выбора тест-кейсов для очередного этапа регрессионного тестирования в условиях острой ограниченности временных ресурсов.</p> <p>22. Что такое модульное тестирование? Какова его цель?</p> <p>23. Перечислите основные возможности модульного тестирования.</p> <p>24. Перечислите основные недостатки модульного тестирования. В каких случаях не следует его применять?</p> <p>25. Что такое методика TDD?</p> <p>26. Опишите основные этапы цикла TDD.</p> <p>27. Что такое рефакторинг?</p> <p>28. Перечислите 3 закона TDD.</p> <p>29. Перечислите основные недостатки TDD.</p> <p>30. Какие существуют варианты организации группы тестирования?</p> <p>31. Перечислите основные задачи группы тестирования в привязке к этапам жизненного цикла проекта.</p> <p>Практические задания</p> <p>1. Пример функциональной спецификации приложения.</p> <p>2. Пример модульного тестирования средствами JUnit.</p> <p>3. Пример интеграционного тестирования средствами JUnit.</p> <p>4. Пример системного тестирования средствами JUnit.</p> <p>5. Пример применения TDD на любом известном языке программирования.</p> <p>Задания на решение задач из профессиональной области, комплексные задания</p> <p><i>Провести комплексное тестирование программного продукта по следующему плану:</i></p> <p>Разработка плана тестирования.</p> <p>Тестирование (инспекция) проектной документации и кода.</p> <p>Реализация модульных тестов, запуск.</p> <p>Реализация интеграционных тестов, запуск.</p> <p>Реализация системных тестов, запуск.</p>

Код индикатора	Индикатор достижения компетенции	Оценочные средства
		<p>Анализ результатов тестирования и подготовка отчета.</p> <p><i>Варианты</i></p> <p>Платформа для удаленной идентификации.</p> <p>Платформа быстрых платежей</p> <p>Платформа-маркетплейс для финансовых услуг и продуктов</p> <p>Платформа для регистрации финансовых сделок</p> <p>Система передачи финансовых сообщений</p> <p>Создание сквозного идентификатора клиента</p> <p>Создание платформы для облачных сервисов</p> <p>Создание платформы на основе технологии распределенных реестров</p> <p>Платформа для электронного взаимодействия.</p>

б) Порядок проведения промежуточной аттестации, показатели и критерии оценивания:

Промежуточная аттестация включает теоретические вопросы, позволяющие оценить уровень усвоения обучающимися знаний, и практические задания, выявляющие степень сформированности умений и владений, проводится в форме экзамена.

Экзамен по данной дисциплине проводится в устной форме по экзаменационным билетам, каждый из которых включает 2 теоретических вопроса и одно практическое задание.

Показатели и критерии оценивания экзамена:

– на оценку **«отлично»** (5 баллов) – обучающийся демонстрирует высокий уровень сформированности компетенций, всестороннее, систематическое и глубокое знание учебного материала, свободно выполняет практические задания, свободно оперирует знаниями, умениями, применяет их в ситуациях повышенной сложности.

– на оценку **«хорошо»** (4 балла) – обучающийся демонстрирует средний уровень сформированности компетенций: основные знания, умения освоены, но допускаются незначительные ошибки, неточности, затруднения при аналитических операциях, переносе знаний и умений на новые, нестандартные ситуации.

– на оценку **«удовлетворительно»** (3 балла) – обучающийся демонстрирует пороговый уровень сформированности компетенций: в ходе контрольных мероприятий допускаются ошибки, проявляется отсутствие отдельных знаний, умений, навыков, обучающийся испытывает значительные затруднения при оперировании знаниями и умениями при их переносе на новые ситуации.

– на оценку **«неудовлетворительно»** (2 балла) – обучающийся демонстрирует знания не более 20% теоретического материала, допускает существенные ошибки, не может показать интеллектуальные навыки решения простых задач.

– на оценку **«неудовлетворительно»** (1 балл) – обучающийся не может показать знания на уровне воспроизведения и объяснения информации, не может показать интеллектуальные навыки решения простых задач.