

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

ТЕСТИРОВАНИЕ И ОТЛАДКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Методические указания к лабораторным занятиям

для направления подготовки:

09.03.04 Программная инженерия

Ставрополь, 2026

Н 63 Тестирование и отладка программного обеспечения: методические указания к лабораторным занятиям для студентов направления 09.03.04 Программная инженерия / Николаев Е.И. – Ставрополь: Изд-во СКФУ, 2026. – 27 с.

Автор:

канд. техн. наук, доцент **Е.И. Николаев**

ОГЛАВЛЕНИЕ

1. Методика и порядок выполнения лабораторных работ	4
2. Краткое теоретическое обоснование.....	7
Лабораторная работа 1. Тестирование черного и белого ящика.....	12
Лабораторная работа 2. Ручное и автоматизированное тестирование	13
Лабораторная работа 3. Функциональное тестирование	14
Лабораторная работа 4. Нефункциональное тестирование	15
Лабораторная работа 5. Unit-тестирование	16
Лабораторная работа 6. Тестирование по типу приложений	17
Список литературы	18

1. МЕТОДИКА И ПОРЯДОК ВЫПОЛНЕНИЯ ЛАБОРАТОРНЫХ РАБОТ

1.1. Решение прикладных задач с использованием шаблонов проектирования

Каждая лабораторная предполагает исследование определенного метода тестирования: описание теоретических принципов метода, описание применения метода тестирования к конкретному программному продукту.

1.2. Порядок защиты

Студент демонстрирует готовый, работающий проект приложения, предоставляет диаграмму классов (и прочие визуальные материалы в нотации UML), демонстрирующую факт наличия разработанного приложения. Затем в соответствии с тематикой лабораторной работы реализуется метод тестирования для приложения.

Также предоставляется отчет по лабораторной работе (*.docx, *.pdf). Рекомендуется сформировать репозиторий GitHub с исходными кодами лабораторных работ.

1.3. Форма отчета

Студент предоставляет отчет в электронном виде. Отчет должен содержать следующие элементы:

1. Титульный лист.
2. Исходный код (в случае объемного материала – ссылка на GitHub).
3. Диаграммы классов для описания приложения.
4. Теоретическое описание метода тестирования.
5. Описание стратегии тестирования конкретного приложения с конкретным методом тестирования.

6. Коды, текстовые материалы, протоколы, входные и выходные данные, используемые в процессе тестирования.

1.4. Варианты заданий

Примерный перечень прикладных задач (студент может самостоятельно предложить и обосновать свою прикладную задачу) для реализации методов тестирования:

1. Разработка приложения «Интерактивная карта»
2. Разработка приложения «Менеджер фотографий»
3. Разработка приложения «Траектория движения»
4. Разработка приложения «Игра «Собиратель монет»
5. Разработка приложения «Свертка изображения»
6. Разработка приложения «Матричный калькулятор»
7. Разработка приложения «Аквалайзер»
8. Разработка приложения «Графический пароль»
9. Разработка приложения «Игра «Танки»
10. Разработка приложения «Игра «Змейка»
11. Разработка приложения «Расчет стоимости поездки»
12. Разработка приложения «Расчет баллистической траектории»
13. Разработка приложения «Интерактивный анализ набора данных»
14. Разработка приложения «Моделирование популяции»
15. Разработка приложения «Обработка изображения фильтрами»
16. Разработка приложения «Игра «Цветы против грибов»
17. Разработка приложения «Фракталы»
18. Разработка приложения «Искусственный интеллект»
19. Разработка приложения «Филиалы и сотрудники»
20. Разработка приложения «Игра «Ракета»
21. Разработка приложения "Игра «Блиц»
22. Разработка приложения «Поиск текста в документах»

- 23. Разработка приложения «Анализ матричных данных»
- 24. Разработка приложения «Игра «Поиск крота»
- 25. Разработка приложения «Вычисления со скобками»

Вариант студента может меняться при выполнении различных лабораторных работ.

2. КРАТКОЕ ТЕОРЕТИЧЕСКОЕ ОБОСНОВАНИЕ

2.1. Понятие тестирования

Тестирование программного обеспечения – это процесс оценки качества разрабатываемого продукта с целью проверки его соответствия заявленным требованиям, надёжности работы в различных условиях и степени удобства для конечного пользователя. Основной задачей тестирования является выявление ошибок, дефектов или несоответствий, которые могут повлиять на функциональность или пользовательский опыт.

Тестирование позволяет обеспечить стабильную и качественную работу программного обеспечения, предотвращая негативный пользовательский опыт и обеспечивая соответствие продукта ожиданиям. Этот процесс необходим для своевременного выявления ошибок, улучшения взаимодействия пользователей с системой и повышения общей надёжности продукта. Среди ключевых преимуществ можно выделить:

1. Выявить ошибки до релиза продукта. Обнаружение багов и сбоев на ранних этапах разработки существенно сокращает стоимость их устранения, избегая влияния на конечных пользователей. Исправление ошибки до релиза обходится в несколько раз дешевле, чем устранение её последствий после выпуска продукта.

2. Гарантировать соответствие требованиям. Этот процесс помогает убедиться, что все заявленные функции работают корректно и соответствуют спецификациям, что позволяет избежать разногласий с клиентами и снизить риск необходимости переработок.

3. Обеспечить конкурентное преимущество. Высокое качество программного обеспечения повышает доверие пользователей, увеличивает их лояльность и снижает вероятность перехода на продукты конкурентов. Пользователи остаются верны бренду, если их ожидания от использования продукта оправдываются.

4. Снизить расходы на поддержку. Устранение дефектов на этапе тестирования минимизирует затраты на последующую поддержку и исправление ошибок, которые могли бы быть обнаружены в процессе эксплуатации. Чем меньше ошибок остаётся в системе после её внедрения, тем ниже расходы на её обслуживание.

2.2. Виды тестирования ПО

Тип тестирования (test type): совокупность тестирующих действий, которая фокусируется на определенных показателях качества. (ГОСТ 56920):

1. Функциональные виды («Что?» – проверяет весь функционал продукта):
 - Функциональное тестирование (Functional testing);
 - Тестирование взаимодействия (Interoperability testing).
2. Нефункциональное («Как?»):
 - Производительности (Performance):
 - 1) тестирование емкости (Capacity testing);
 - 2) нагрузочное (Load testing);
 - 3) стрессовое (Stress testing);
 - 4) масштабируемости (Scalability test);
 - 5) объемное тестирование (Volume testing);
 - 6) выносливости (Soak/Endurance testing);
 - 7) устойчивости (Resilience testing);
 - 8) стабильности/надежности (Stability / Reliability testing);
 - 9) отказ и восстановление (Failover and Recovery testing);
 - 10) эталонное и тестирование базовой версии (Benchmark and Baseline Testing).
 - Тестирование безопасности (Security and Access Control testing);
 - Удобство пользования (Usability testing);
 - Тестирование доступности (Accessibility testing);
 - Тестирование установки (Installation testing);

- Тестирование на соответствие (Conformance/Compliance testing);
- Конфигурационное (Configuration testing);
- Тестирование локализации, глобализации и интернационализации.

3. Связанное с изменениями:

- Регрессионное (Regression testing);
- Тест работоспособности (Sanity testing);
- Дымовое (Smoke testing);

Это не исчерпывающая классификация методов тестирования. На рис. 1.1 показана таксономия подходов к тестированию приложений.

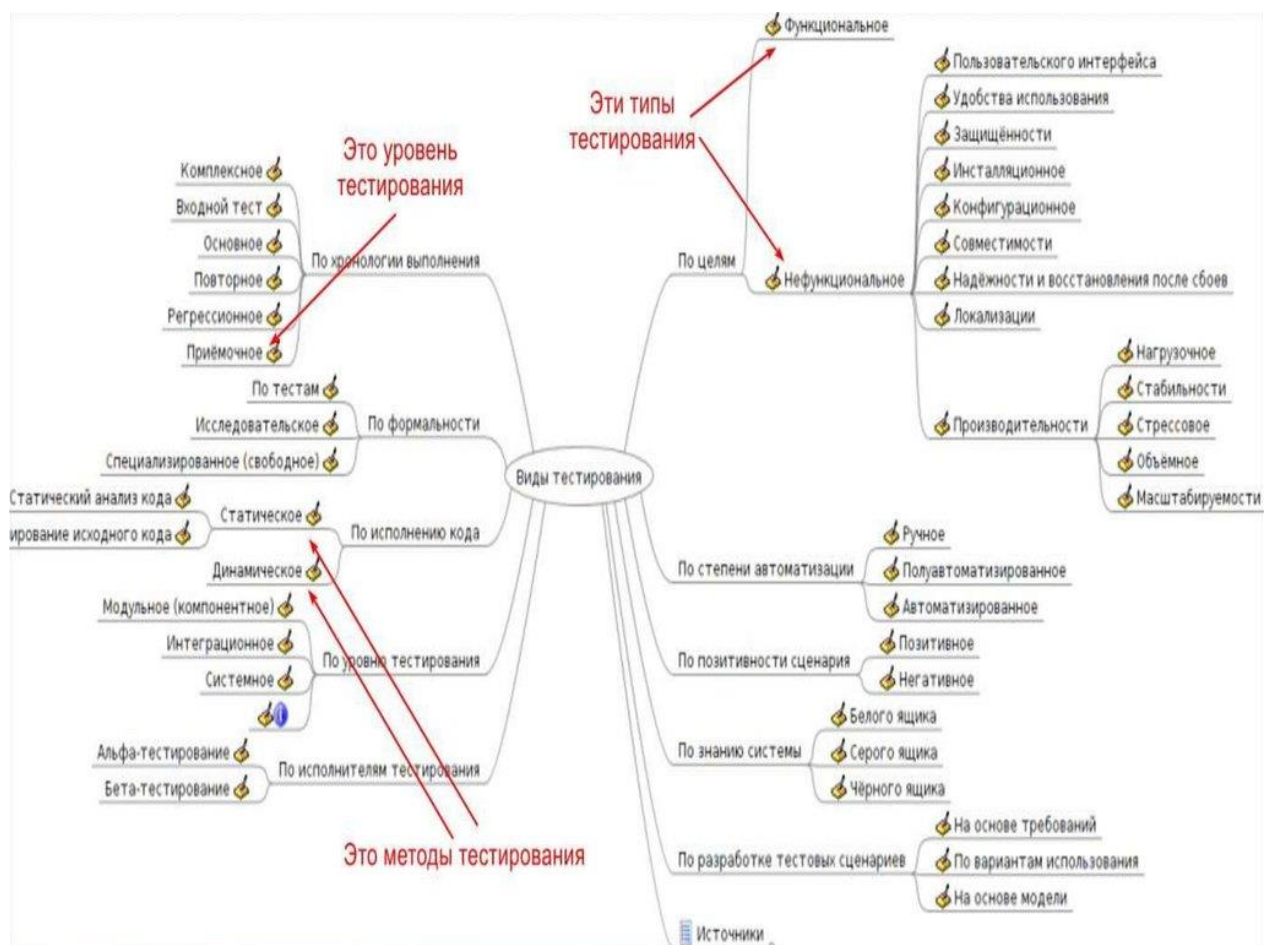


Рисунок 1.1 – Классификация тестирования: отображены методы, типы и уровни тестирования.

На рис. 1.2 показана еще одна попытка классифицировать подходы к тестированию.

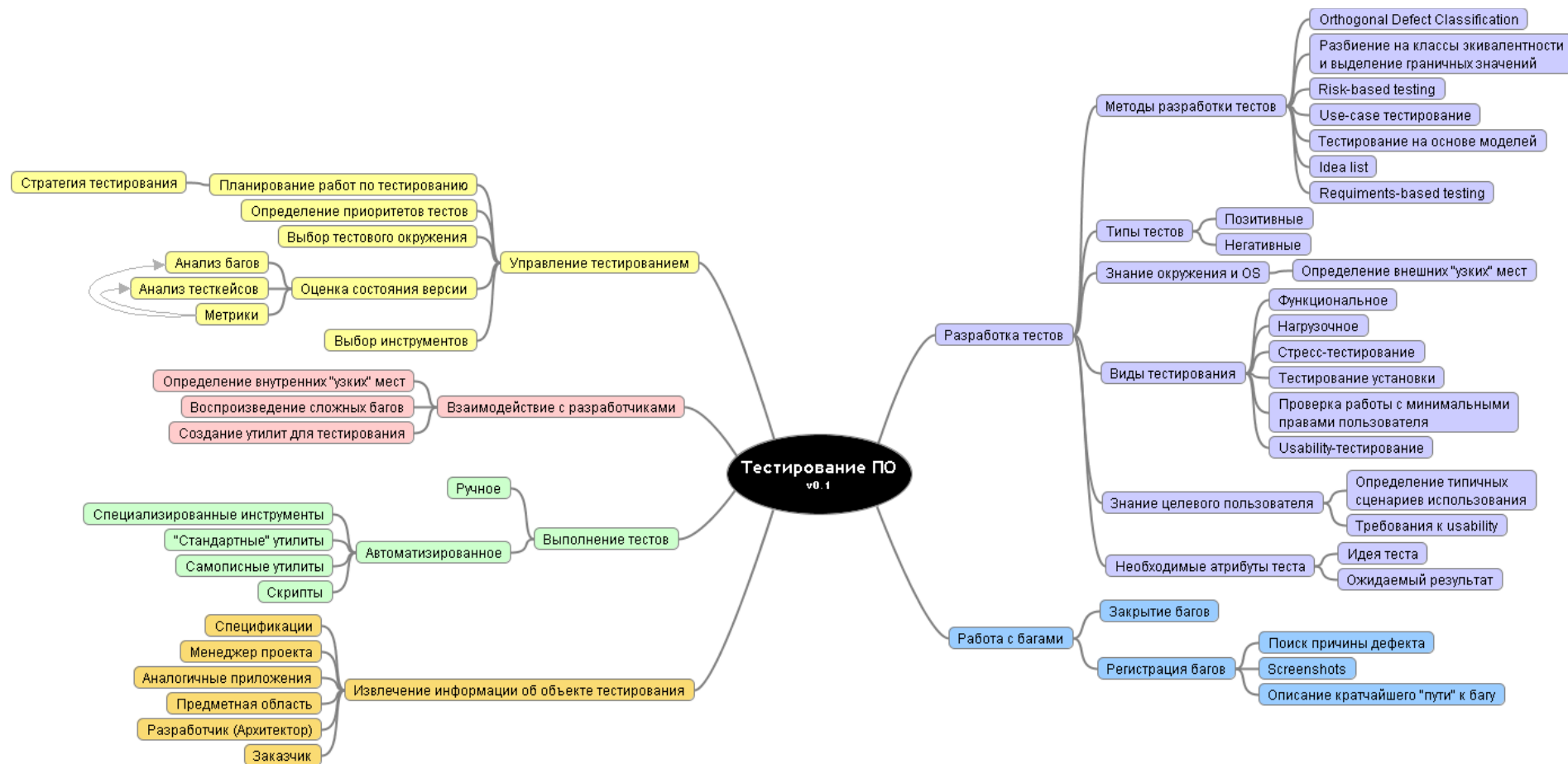


Рисунок 1.2 – Классификация тестирования.

ЛАБОРАТОРНАЯ РАБОТА 1. ТЕСТИРОВАНИЕ ЧЕРНОГО И БЕЛОГО ЯЩИКА

Задача

Для разработанного приложения (пп. 1.4) продемонстрируйте на выбор применение технологии тестирования «черного ящика» или «белого ящика».

ЛАБОРАТОРНАЯ РАБОТА 2. РУЧНОЕ И АВТОМАТИЗИРОВАННОЕ ТЕСТИРОВАНИЕ

Задача

Для разработанного приложения (пп. 1.4) продемонстрируйте на выбор применение технологии тестирования: ручное или автоматизированное.

ЛАБОРАТОРНАЯ РАБОТА 3. ФУНКЦИОНАЛЬНОЕ ТЕСТИРОВАНИЕ

Задача

Для разработанного приложения (пп. 1.4) продемонстрируйте на выбор применение технологии функционального тестирования:

Компонентное (модульное) тестирование – проверка отдельных единиц кода на корректность.

Интеграционное тестирование – проверка правильности взаимодействия между компонентами.

Системное тестирование – комплексная проверка всей системы на соответствие требованиям.

Приемочное тестирование – подтверждение готовности продукта к передаче заказчику.

Регрессионное тестирование – проверка, что новые изменения не повредили существующую функциональность.

Smoke-тестирование – быстрая проверка критических функций после сборки.

Следует учитывать, что процесс функционального тестирования включает следующие шаги:

- Анализ требований и спецификаций
- Создание тестовых сценариев на основе функциональных требований
- Подготовка тестовых данных
- Выполнение тестов и сравнение фактических результатов с ожидаемыми
- Документирование и отслеживание обнаруженных дефектов
- Повторное тестирование после исправления дефектов

ЛАБОРАТОРНАЯ РАБОТА 4. НЕФУНКЦИОНАЛЬНОЕ ТЕСТИРОВАНИЕ

Задача

Для разработанного приложения (пп. 1.4) продемонстрируйте на выбор применение технологии нефункционального тестирования:

Тестирование производительности – оценка быстродействия, отзывчивости и стабильности системы под нагрузкой.

Нагрузочное тестирование – проверка поведения при ожидаемой рабочей нагрузке.

Стресс-тестирование – определение пределов работоспособности системы.

Тестирование безопасности – выявление уязвимостей и проблем с защитой данных.

Тестирование удобства использования – оценка дружелюбности интерфейса и пользовательского опыта.

Тестирование совместимости – проверка корректной работы на различных платформах и устройствах.

ЛАБОРАТОРНАЯ РАБОТА 5. UNIT-ТЕСТИРОВАНИЕ**Задача**

Для разработанного приложения (пп. 1.4) продемонстрируйте применение Unit-тестирования.

ЛАБОРАТОРНАЯ РАБОТА 6. ТЕСТИРОВАНИЕ ПО ТИПУ ПРИЛОЖЕНИЙ

Задача

Для разработанного приложения (пп. 1.4) продемонстрируйте применение методологии тестирования из списка:

Тестирование веб-приложения.

Тестирование мобильного приложения.

Тестирование десктоп-приложения.

Выбирать метод необходимо в соответствии с приложением, которое подвергается тестированию.

СПИСОК ЛИТЕРАТУРЫ

Список основной литературы

1. Канер, С. Тестирование программного обеспечения : фундаментальные концепции менеджмента бизнес-приложений / С. Канер, Дж. Фолк, Е. К. Нгуен ; перевод с английского под редакцией А. М. Смирнова. – Москва : Издательство И. В. Балабанова, 2021. – 544 с. – ISBN 978-5-901015-21-4. – Текст : непосредственный.
2. Куликов, С. С. Тестирование программного обеспечения : базовый курс / С. С. Куликов. – 3-е изд. – Минск : Четыре четверти, 2020. – 312 с. – ISBN 978-985-581-419-2. – Текст : непосредственный.
3. Майерс, Г. Искусство тестирования программ / Г. Майерс, Т. Баджетт, К. Сандлер. – 3-е изд. – Санкт-Петербург : Диалектика, 2020. – 272 с. – ISBN 978-5-907203-66-2. – Текст : непосредственный.
4. Матвеева, Н. Шпаргалка начинающего тестировщика / Н. Матвеева. – Санкт-Петербург : Питер, 2024. – 256 с. – ISBN 978-5-4461-2292-9. – Текст : непосредственный.
5. Назина, О. Что такое тестирование: курс молодого бойца / О. Назина. – Санкт-Петербург : БХВ-Петербург, 2022. – 320 с. – ISBN 978-5-9775-6831-9. – Текст : непосредственный.
6. Савин, Р. Тестирование dot com, или Пособие по жестокому обращению с багами в интернет-стартапах / Р. Савин. – Москва : Дело, 2022. – 316 с. – ISBN 978-5-7749-1412-2. – Текст : непосредственный.
7. Хориков, В. Принципы юнит-тестирования / В. Хориков. – Санкт-Петербург : Питер, 2021. – 320 с. – ISBN 978-5-4461-1718-5. – Текст : непосредственный.

Список дополнительной литературы

8. Black, R. Critical Testing Processes: Plan, Prepare, Perform, Perfect / R. Black. – Boston : Addison-Wesley, 2021. – 576 p. – ISBN 978-0-201-74017-2. – Text : unmediated.

9. Copeland, L. A Practitioner's Guide to Software Test Design / L. Copeland. – Boston : Artech House, 2019. – 320 p. – ISBN 978-1-58053-791-9. – Text : unmediated.

10. Whittaker, J. A. How Google Tests Software / J. A. Whittaker, J. Arbon, J. Carollo. – Upper Saddle River : Addison-Wesley, 2012. – 288 p. – ISBN 978-0-321-80302-3. – Text : unmediated.

Перечень Интернет-ресурсов

1. <https://habr.com/ru/articles/587620/> – Просто о тестировании.
2. <https://habr.com/ru/companies/otus/articles/923480/> – описание методов тестирования.

