

72. ОБЗОР МЕТОДОВ ТЕСТИРОВАНИЯ ПРИ РАЗРАБОТКЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Гутник Э. А., Ермантович М. А. студенты группы 272301, Полоско Е.И., аспирант,

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Ефремов А.А. – канд. экон. наук, доцент каф. ЭИ

Аннотация. Рассматриваются современные методы тестирования в контексте разработки программного обеспечения. Уделяется внимание описанию различных подходов к тестированию. Проанализированы преимущества и недостатки каждого метода, а также рассмотрены современные тенденции и инновации в области тестирования программного обеспечения.

Разработка программного обеспечения является сложным процессом, требующим не только глубоких знаний в области программирования, но и эффективных методов тестирования. Тестирование программного обеспечения – это аналитический процесс, направленный на оценку соответствия разработанного продукта установленным стандартам качества [1]. В современном мире существует множество методов тестирования, каждый из которых имеет свои преимущества и недостатки.

Уровни тестирования программного обеспечения:

1. Модульное тестирование – это процесс проверки отдельных модулей или компонентов программы. Целью модульного тестирования является проверка корректности работы каждого модуля независимо от других частей программы.

2. Интеграционное тестирование – это процесс проверки взаимодействия между различными модулями или компонентами программы после их объединения. Целью этого вида тестирования является обнаружение ошибок, связанных с передачей данных между модулями, интеграцией различных частей системы и общей работоспособностью программы.

3. Системное тестирование – это проверка всей системы в ее целостности. В отличие от интеграционного тестирования, которое сосредотачивается на проверке отдельных компонентов, системное тестирование оценивает работу системы в целом. Этот вид тестирования включает в себя проверку соответствия программы требованиям заказчика, ее производительности, безопасности и удобства использования.

4. Приемочное тестирование – это процесс проверки системы на соответствие требованиям заказчика. Ключевой особенностью является утверждение требований. Осуществляется проверка, соответствуют ли результаты системы ожиданиям и требованиям заказчика [2].



Рисунок 1.1 – Уровни тестирования ПО

Каждый из этих видов тестирования играет важную роль в обеспечении качества программного обеспечения. Модульное тестирование гарантирует корректность отдельных компонентов, интеграционное – их взаимодействие, системное – работоспособность системы в целом, а приемочное – удовлетворение требований заказчика. Вместе они составляют комплексный подход к тестированию, обеспечивающий надежность и качество программного продукта.

По степени автоматизации выделяют ручное и автоматизированное тестирование. Главным различием между этими методами тестирования ПО является то, что в ручном тестировании тест-кейсы выполняет только человек, а в автоматизированном частично или полностью специальное инструментальное средство.

По принципам работы с ПО выделяют два вида тестирования, позитивное и негативное. Позитивное тестирование – это тестирование, где все действия с приложением выполняются строго по инструкции без исключительных ситуаций (иногда говорят тестирование в «тепличных условиях»). Негативное тестирование – при выполнении тестирования с приложением выполняются некорректные операции, а также применяются данные, приводящие к ошибкам [3].

Существует три метода тестирования ПО: «белый ящик», «чёрный ящик» и «серый ящик».

«Белый ящик», или структурное тестирование, основывается на знании внутренней структуры программы. Тестировщики имеют доступ к исходному коду и используют его для создания тестовых случаев. Они проверяют каждую логическую ветвь, путь выполнения и участки кода на предмет ошибок.

Преимущества белого ящика:

- глубокое понимание кода: благодаря доступу к коду, тестировщики могут лучше понять внутреннюю структуру приложения и его алгоритмы, что позволяет им выявлять скрытые ошибки;
- раннее обнаружение проблем: белый ящик может быть использован на ранних этапах разработки для быстрого выявления проблем и исправления их до того, как они превратятся в критические ошибки.

Недостатки белого ящика:

- требует доступ к коду: для проведения белого ящика необходимо иметь доступ к исходному коду, что может быть проблематично при тестировании сторонних программ или коммерческого ПО;
- ориентирован на код: белый ящик фокусируется исключительно на коде, что может упустить некоторые аспекты функциональности приложения.

«Чёрный ящик», или функциональное тестирование, основывается на внешнем поведении программы без знания ее внутренней структуры. Тестировщики создают тестовые случаи на основе спецификаций, требований и ожидаемого поведения приложения.

Преимущества черного ящика:

- независимость от кода: тестировщики могут работать независимо от разработчиков, так как им не требуется доступ к исходному коду;
- ориентирован на пользователей: черный ящик тестирует приложение с точки зрения конечного пользователя, что помогает выявить проблемы, связанные с удобством использования и интерфейсом.

Недостатки черного ящика:

- недостаточное покрытие кода: поскольку тестировщики не имеют доступа к исходному коду, некоторые участки кода могут остаться непроверенными;
- сложность выявления скрытых ошибок: некоторые ошибки, связанные с внутренними аспектами приложения, могут быть трудно обнаружить через черный ящик.

«Серый ящик» – это комбинация белого и черного ящиков. Тестировщики имеют частичное знание о внутренней структуре приложения, что помогает им создавать более эффективные тесты, ориентированные на важные аспекты кода и функциональности.

*60-я Юбилейная Научная Конференция Аспирантов, Магистрантов и Студентов БГУИР,
Минск 2024*

Преимущества серого ящика:

– баланс между покрытием кода и функциональностью: серый ящик обеспечивает более широкое покрытие кода, чем черный ящик, и в то же время сохраняет независимость от кода;

– более эффективное обнаружение ошибок: благодаря частичному знанию о внутренней структуре приложения, тестировщики могут создавать более целенаправленные тесты, что помогает выявлять скрытые ошибки.

Недостатки серого ящика:

– требует дополнительных знаний: для проведения серого ящика тестировщики должны иметь достаточное понимание внутренней структуры приложения, что может потребовать дополнительных усилий и времени;

– сложнее настроить: создание тестов для серого ящика может потребовать больше усилий и ресурсов, чем для черного ящика, из-за необходимости сочетать знание внутренней структуры с внешним поведением приложения.

Современные тенденции в тестировании ПО включают автоматизацию, интеграцию с DevOps и непрерывное тестирование, использование облачных платформ, применение искусственного интеллекта и машинного обучения, уделяя внимание тестированию мобильных приложений, безопасности и соответствию [4]. Эти инновации помогают улучшить эффективность, качество и безопасность программного обеспечения.

Список использованных источников:

1. Информационные технологии и управление: материалы 49 науч. конф. аспирантов, магистрантов и студентов, Минск, 6–10 мая 2013 г. / Белорус. гос. ун-т информатики и радиоэлектроники; редкол.: Л. Ю. Шилин [и др.]. – Минск: БГУИР, 2013. – 103 с.

2. Инженерное образование в цифровом обществе: материалы Междунар. науч.-метод. конф. (Республика Беларусь, Минск, 14 марта 2024 года). В 2 ч. Ч. 1 / редкол.: Е. Н. Шнейдеров [и др.]. – Минск: БГУИР, 2024. – 372 с.

3. Тестирование программного обеспечения: учебное пособие // С. С. Куликов – базовый курс 3-е издание, 2023. – 68 с.

4. Шевчук В.И. Современные подходы к тестированию программного обеспечения // Шевчук В. И // Электронный ресурс, Режим доступа: <https://vc.ru/u/1205966-vital-shauchuk/691440-sovremennye-podhody-k-testirovaniyu-programmnogo-obespecheniya>