

102. ТЕСТИРОВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ В ПРОЦЕССЕ РАЗРАБОТКИ ПРИКЛАДНЫХ ПРОГРАММ

Барадулькина А.С., студент гр. 373904, Федюкович Т.В., ассистент кафедры ЭИ

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

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

Аннотация: в работе рассматривается важность тестирования программного обеспечения, его основные преимущества. Объясняется принцип тестирования «белого ящика» и «чёрного ящика». Описываются уровни тестирования и их практическое применение.

Ключевые слова: тестирование, программное обеспечение, разработка программного обеспечения, методика и принципы тестирования.

Любая IT-компания, разрабатывающая и поставляющая на рынок свой продукт, должна принимать во внимание, что тестирование программного обеспечения является важным и ценным этапом жизненного цикла его разработки. В случае, когда компании игнорируют процесс тестирования, страдает как сам продукт, так и бизнес компании-разработчика в целом, поскольку ошибки в программном обеспечении могут быть дорогостоящими или даже опасными. Последнее десятилетие в IT-индустрии интенсивно ведутся исследования в области разработки методик тестирования [4], позволяющих повысить эффективность процессов управления жизненным циклом программного продукта. Тестирование программного обеспечения в процессе разработки прикладных программ имеет следующие преимущества.

Тестирование в проекте может сэкономить деньги в долгосрочной перспективе. Разработка программного обеспечения состоит из множества этапов, и, если ошибки обнаруживаются на более ранних этапах, их исправление обходится намного дешевле. Чем раньше будут найдены ошибки, тем ниже затраты на их исправление и риски по срыву сроков реализации. Значит, имеет смысл привлекать тестировщиков к проверкам как можно раньше.

Существует много прецедентов, когда пользовательские данные были украдены или злоумышленники использовали их в своих интересах. Люди хотят использовать надежный и

безопасный продукт. Личная информация пользователей должна оставаться как можно в большей конфиденциальности, особенно при использовании услуг, где она является уязвимой для пользователей, например, банковская информация, данные безопасности и т.д. Тестирование же позволяет устранить риски и снизить проблемы уязвимости конфиденциальных данных на раннем этапе.

Продукты всегда определенным образом служат пользователям, поэтому очень важно, чтобы они соответствовали их ожиданиям и запросам, а также могли надежно реализовать все заявленные функции, и, следовательно, обеспечить отличное обслуживание клиентов. Даже после того, как разработчики устранят явные проблемы, может «неожиданно» появиться другая проблема, как следствие несогласованности в процедурах контроля функционала, из-за возможных ошибок сборки версий при коллективной разработке сложных программных продуктов. Для решения этой задачи в компаниях-разработчиках программного обеспечения необходимо внедрять согласованные стратегии тестирования, которые включают модульный подход, интеграцию, системное тестирование и другие методы, обычно входящие в систему управления разработки продукта. Компании, занимающиеся тестированием, обычно располагают множеством устройств и инструментов, доступных для охвата как можно большего количества сценариев тестирования. Компания, чей продукт или услуга имеют хорошее качество, чей сайт или приложение хорошо работают, позиционируется на рынке как надежная и уважаемая. Это более привлекательно для клиентов и партнеров, так что это дает потенциал для получения еще большего дохода от своего продукта [1].

Современные методологии разработки предполагают инкрементное тестирование, то есть возможность внесения необходимых коррекций по мере совершенствования версий продукта, начиная с этапа уточнения требований к продукту и составления технического задания. Методики тестирования включают процедуры анализа с применением методологии моделирования [4]. На этапе разработки модели можно максимально эффективно протестировать степень удовлетворенности клиента функционалом продукта и его требований к интерфейсу. Для решения этой задачи необходимо подключать к процессам тестирования клиентов, для расширения класса решаемых задач и выявления насущных потребностей конечного пользователя. При использовании такого подхода специалист-тестировщик получает более широкую картину для анализа, за счет вовлечения его в процесс коллективной разработки.

Тесное сотрудничество тестировщиков с разработчиками улучшает их понимание каждой части кода. Их взаимодействие через обмен данными и сочетание процедур анализа, проектирования и тестирования позволяет систематизировать контроль качества, сокращая сроки разработки продукта и снижая стоимость внесения исправлений. Более тесное взаимодействие разработчиками позволяет тестировщикам сконцентрироваться на тех областях, где наиболее высок риск возникновения ошибок, и соответствующим образом выстроить приоритеты. В свою очередь, обмен данными полезен и разработчикам для скорейшего управления дефектами и снятия рисков [1].

Для оптимального выбора конкретной методики тестирования должно учитываться множество специфических факторов проекта разработки: масштаб продукта и его сложность, ресурсные ограничения, особенности инфраструктуры, квалификацию персонала и ряд других. Следует отметить, что первоочередным вопросом в определении стратегии нужно сделать выбор базового этапа жизненного цикла разработки для основного тестирования. Также необходимо вводить процедуры тестирования в графики сетевого планирования для учета временных затрат и реалистичных оценок сроков завершения продукта, ведь как можно видеть из, например, отчетов CHAOS Report от The Standish Group, превышение относительно ожидаемого времени реализации разработки характерно для 45% ИТ-проектов. Далее необходимо выявить набор стабильных однозначных параметров, характеризующих успешность процедуры тестирования [3].

Выделяют следующие уровни тестирования программного обеспечения:

1. Тестирование компонентов – установление источников дефектов и погрешностей в компонентах программы (например, отдельный класс или функция), устранение первичных ошибок (исходя из их вторичных проявлений) [2].

2. Интеграционное тестирование – отдельные программные модули объединяются и тестируются в группе (например, как база данных взаимодействует со степенью абстракции информации).

3. Системное тестирование – это тестирование производится на интегрированной системе с целью проверки соответствия системы с исходными требованиями. Имеет подкатегории:

На этапе тестирования программного обеспечения тестировщик рассматривает вероятность обнаружения всех ошибок программ. Для изучения данного вопроса рассмотрим исследования стратегий тестирования [4].

Стратегия «белого ящика» исследует внутреннее строение программы, т.е. тестировщик получает тестовые показатели посредством проверки логики программы. Текущий метод тестирования не способен определить неосуществленные компоненты спецификации, дефицит условий или

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

формирование не того приложения. Эта стратегия имеет несколько критериев покрытия кода: покрытие операторов, условий, функций, путей, значений параметров.

Стратегия «черного ящика» представляет отладку эргономичного поведения объекта с точки зрения внешнего мира, где не требуется наличие знания о внутреннем устройстве. Задачей данного тестирования становится обнаружение обстоятельств, при которых исполнение программы не соответствует ее условиям. В таком случае выявление всех погрешностей в программе идентифицируется показателем полного входного тестирования (это может быть достигнуто только при условии эксплуатации все возможных наборов входных данных).

Сравнивая обе стратегии, можно сделать заключение, что они не могут быть максимально полезными и эффективными при тестировании программного обеспечения. В связи с этим большая часть специалистов реализуют два метода в совокупности друг с другом. Это называется стратегией «серого ящика» [4].

Таким образом, программный продукт не может быть завершенным и пригодным к использованию без прохождения тестирования, так как оно позволяет избавиться от большинства ошибок в программе, повышая качество программного обеспечения.

Список использованных источников:

1. Канер С., Фолк Д., Нгуен К. Е. Тестирование программного обеспечения. Фундаментальные концепции менеджмент бизнес- приложений. ДиаСофт, 2001г., 544 стр.
2. Липаев В.В. Тестирование компонентов и комплексов программ. М: СИНТЕГ, 2010.
3. Логачева Н. В., Ладоньчева М. Л., Пузырева К. С. Важность тестирования программного обеспечения в процессе разработки программного обеспечения //Инновационная наука. – 2022. – №. 2-2. – С. 23-26.
4. Степанченко И.В. Методы тестирования программного обеспечения. Волгоград, : РПК «Политехник», 2006.