

УДК 004.223.2

ТЕСТИРОВАНИЕ ПРОГРАММНОГО ПРОДУКТА

К.В. СТАНКЕВИЧ

(Представлено: канд. физ.-мат. наук., доц. Ю.Ф. ПАСТУХОВ)

В статье рассматриваются различные виды тестирования. Проведены исследования по использованию различных видов тестирования на разных этапах разработки программного обеспечения.

Ключевые слова: информационные технологии, тестирование ПО, информационные системы, информация, функциональное тестирование, регрессионное тестирование, функциональное тестирование, нефункциональное тестирование.

Тестирование – ряд мероприятий, связанных с различного рода испытаниями объекта тестирования с целью установления соответствия или несоответствия его характеристик определенным требованиям и выявления дефектов. Дефектами, в свою очередь, могут быть как ошибки в работе, так и неприемлемое качество функционирования в определенных условиях эксплуатации.

Результаты тестирования - это список выявленных несоответствий и дефектов, но, как правило, без указания их причин, хотя, опытный специалист по качеству продукции в состоянии провести первичную диагностику некорректного поведения объекта и сообщить диагноз его разработчикам.

Тестирование программного продукта проводят как на этапах создания, так и на этапах технического сопровождения и дальнейшего развития.

Тестирование на этапе создания программного продукта. Виды тестирования на этапе создания программного продукта классифицируют по преследуемым этим процессом целям. Функциональное тестирование – проверка корректного выполнения программой заложенных в нее функций. Нефункциональное тестирование – все прочие виды испытаний, такие как тестирование производительности в различных режимах эксплуатации, тестирование эргономики пользовательского интерфейса, тестирование отказоустойчивости и т.д.

Функциональное тестирование. Для проведения функционального тестирования персоналом отдела технического контроля разрабатывается документ программа и методика испытаний функционала приложения (ПМИ). Документ ПМИ содержит перечень сценариев тестирования программного продукта (test cases) с подробным описанием шагов. Каждый шаг сценария тестирования характеризуется действиями пользователя (специалиста по тестированию) и ожидаемыми результатами – ответной реакцией программы на эти действия. Программа и методика испытаний обязана имитировать эксплуатацию программного продукта в реальном режиме. Это означает, что сценарий тестирования должен быть построен на основе анализа операций, которые будут выполнять будущие пользователи системы, а не быть искусственно составленной последовательностью понятных только разработчику манипуляций. Функциональное тестирование может проводиться на различных уровнях тестирования, перечень которых зависит от сложности приложения:

- компонентное (модульное) тестирование,
- интеграционное тестирование.

Каждый компонент системы может быть рассмотрен, как отдельная подсистема, и внутри этой подсистемы могут быть выделены ее компоненты. В этом случае для таких подсистем отдельно проводятся компонентное и интеграционное тестирование. Таким образом, для сложных программных продуктов получаем иерархическую структуру процесса тестирования, на каждом уровне которой объектом тестирования является определенная часть программного комплекса (рисунок).



Рисунок. – Пример иерархической структуры процесса тестирования программного продукта

Выделение уровней может происходить по принципу общей функциональности (подсистема ввода/вывода данных, подсистема расчетов и аналитики, подсистема хранения данных и т.п.), по принадлежности к конкретной части проектного решения (сервер, клиент, посредник), по используемым технологиям, либо по всему сразу. В этом случае тестирование может проводиться снизу-вверх, и при каждом переходе на более высокий уровень, протестированные ранее части программы уже выступают в качестве компонентов подсистемы более высокого порядка. Бывает, что проводят тестирование сверху вниз, начиная от испытаний, связанных с общим поведением программы и заканчивая деталями. В этой ситуации может оказаться так, что на высоких уровнях абстракции реализация деталей еще отсутствует и заменяется различного рода имитаторами – функциональными заглушками, которые только со временем, при переходе на более низкие уровни заменяются реальными функциональными компонентами. Такой процесс на ранних этапах является ничем иным как тестированием прототипа программного продукта.

Нефункциональное тестирование. Нефункциональное тестирование, как следует из самого названия, в первую очередь оценивает второстепенные качества функционирования программного продукта. Второстепенные не в том смысле, что эти качества программного продукта менее важны, чем корректное выполнение им своих основных функций, а в том, что они никому не интересны, если продукт, в принципе, не выполняет то, что должен. Кому, например, интересны результаты замеров производительности подсистемы расчетов, если сами результаты этих расчетов не верны? Для кого имеет значение удобство пользовательского интерфейса программы, если его работа не приводит к ожидаемым от него результатам? Ну а насчет важности нефункционального тестирования можно сказать так: иногда его результаты имеют более серьезный вес в общей оценке работы программного продукта, чем результаты функционального тестирования. Например, для системы, работающей в реальном времени куда как более важно успевать взаимодействовать со всеми наблюдаемыми внешними объектами, чем корректно, с точки зрения функциональных требований, обрабатывать события от конкретного источника. Повторюсь, что это вовсе не означает, что можно некорректно взаимодействовать с отдельно-взятым объектом в угоду производительности.

Тестирование производительности. Достаточно интересные виды тестирования с точки зрения условий их проведения и оценки результатов. В ходе этапа тестирования производительности в первую очередь проводят нагрузочное тестирование, целью которого является проверка, будет ли система адекватно реагировать на внешние воздействия в режиме, близком к режиму реальной эксплуатации. Адекватная реакция – это время отклика в пределах, установленных требованиями к программному продукту. Внешние воздействия (нагрузка) – это, чаще всего, запросы пользователей. Ну а режим реальной эксплуатации подразумевает то, что система развернута на соответствующем требованиям аппаратном обеспечении, а самих пользователей примерно столько, сколько и планировалось. Кроме нагрузочного тестирования проводят испытания в условиях минимальных аппаратных средств и максимальной нагрузки – стрессовое тестирование, а также, испытания в условиях предельных объемов обрабатываемой информации – объемное тестирование. Выделяют еще один вид тестирования: тестирование стабильности и надежности, которое включает в себя не только длительное испытание программного продукта в нормальных условиях, но и способность его возвращаться в нормальный режим функционирования после непродолжительных периодов стрессовых нагрузок.

Прочие виды нефункционального тестирования. К прочим видам тестирования относят тестирование процесса установки или развертывания программного продукта, тестирование эргономики пользовательского интерфейса (не корректной его работы, а именно удобства использования), тестирование способности системы к восстановлению нормальной работы после серьезных сбоев, вызванных отказами аппаратного или системного программного обеспечения. Отдельно проводят испытания системы на различных конфигурациях, если требованиями они предусмотрены. Конфигурации могут отличаться вплоть до операционных систем серверной и клиентской частей программного комплекса. Тестирование отдельной конфигурации может сводиться к проведению для нее всего комплекса испытаний программного продукта.

Тестирование на этапе сопровождения программного продукта.

Регрессионное тестирование. Регрессионное тестирование проводят по результатам исправления выявленных на этапе эксплуатации программного продукта ошибок и дефектов. Также, к этому виду тестирования относят испытания программного продукта после внесения в него незначительных изменений, которые не должны влиять на общую функциональность, а вызваны такими обстоятельствами, как необходимость перехода на новую версию операционной системы или новый набор прикладных библиотек стороннего производителя. Цель регрессионного тестирования проста: доказать, что «ничего не сломалось», и что программный продукт по-прежнему соответствует всем заявленным ранее требованиям.

Предварительное тестирование новой версии программного продукта. Комплекс предварительных мероприятий, направленный на то, чтобы быстро показать жизнеспособность новой версии программного продукта или отправить ее на доработку, выявив в максимально короткие сроки наиболее серьезные дефекты. Ключевое слово в определении – быстро. Подобный вид тестирования еще называют дымовым тестированием по аналогии с испытаниями печниками новых печей: если дым не повалил оттуда, откуда

не положено, значит все в порядке. Если предварительное тестирование заканчивается успешно, новую версию программного продукта отправляют на более детальное «обследование», которое включает функциональное и нефункциональное тестирование.

Заключение. Виды и структура процесса тестирования определяются сложностью самого объекта тестирования. То же самое с теми, кто тестирует программный продукт. Ими могут быть и сами программисты (предварительное тестирование), могут быть сотрудники отдела качества компании разработчика, а могут быть и привлеченные ресурсы. Например, если программа является серийным продуктом, то все ее основные испытания заканчиваются системным тестированием, а потом уже начинаются рекламные кампании и продажи. Если программный продукт – частный заказ, то после системного тестирования проводят еще и приемочное тестирование. И системное тестирование, и приемочное тестирование направлены на комплексное испытание системы (все виды предусмотренного для нее функционального и нефункционального тестирования), но проводятся на разных площадках и с участием разного персонала. Системное тестирование проводится на стороне разработчика, а приемочное на стороне заказчика и на его же аппаратном и системном программном обеспечении. Что касается процесса тестирования на этапе сопровождения, то он может длиться намного дольше, чем сам процесс создания программного решения и прерываться только выпуском абсолютно новой версии программного продукта, после чего все начинается заново.

ЛИТЕРАТУРА

1. Тестирование и качество ПО [Электронный ресурс] – Режим доступа: <https://software-testing.ru/> – Дата доступа: 23.09.2019.
2. Виды Тестирования ПО [Электронный ресурс] – Режим доступа: <https://www.it-courses.by/all-software-testing-types/> – Дата доступа: 23.09.2019.