

ИССЛЕДОВАНИЕ МЕТОДОВ ТЕСТИРОВАНИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ПО УРОВНЮ, ОБЪЕКТАМ И АВТОМАТИЗАЦИИ ТЕСТИРОВАНИЯ

Тюленева Ирина Анатольевна

магистрант, Московский государственный технологический университет СТАНКИН, РФ, г. Москва

Новоселова Ольга Вячеславовна

канд. техн. наук, доцент, Московский государственный технологический университет СТАНКИН, РФ, г. Москва

Качество программного обеспечения (ПО) напрямую зависит от соответствия результата ожиданиям заказчика. Таким образом, главной задачей при разработке программного продукта является выявление требований и критериев, которые обеспечили бы ожидаемое качество ПО. Для определения степени соответствия полученного программного обеспечения с ожидаемым, необходимо провести различные мероприятия по тестированию разработанного программного обеспечения. Это дает возможность обнаружить ряд ошибок, устранение которых позволит добиться необходимого качества продукта.

Существует множество видов тестирования, классификация которых условно разделена по группам, в зависимости от того, на какие цели опирается определенное тестирование программного обеспечения (Рис. 1).

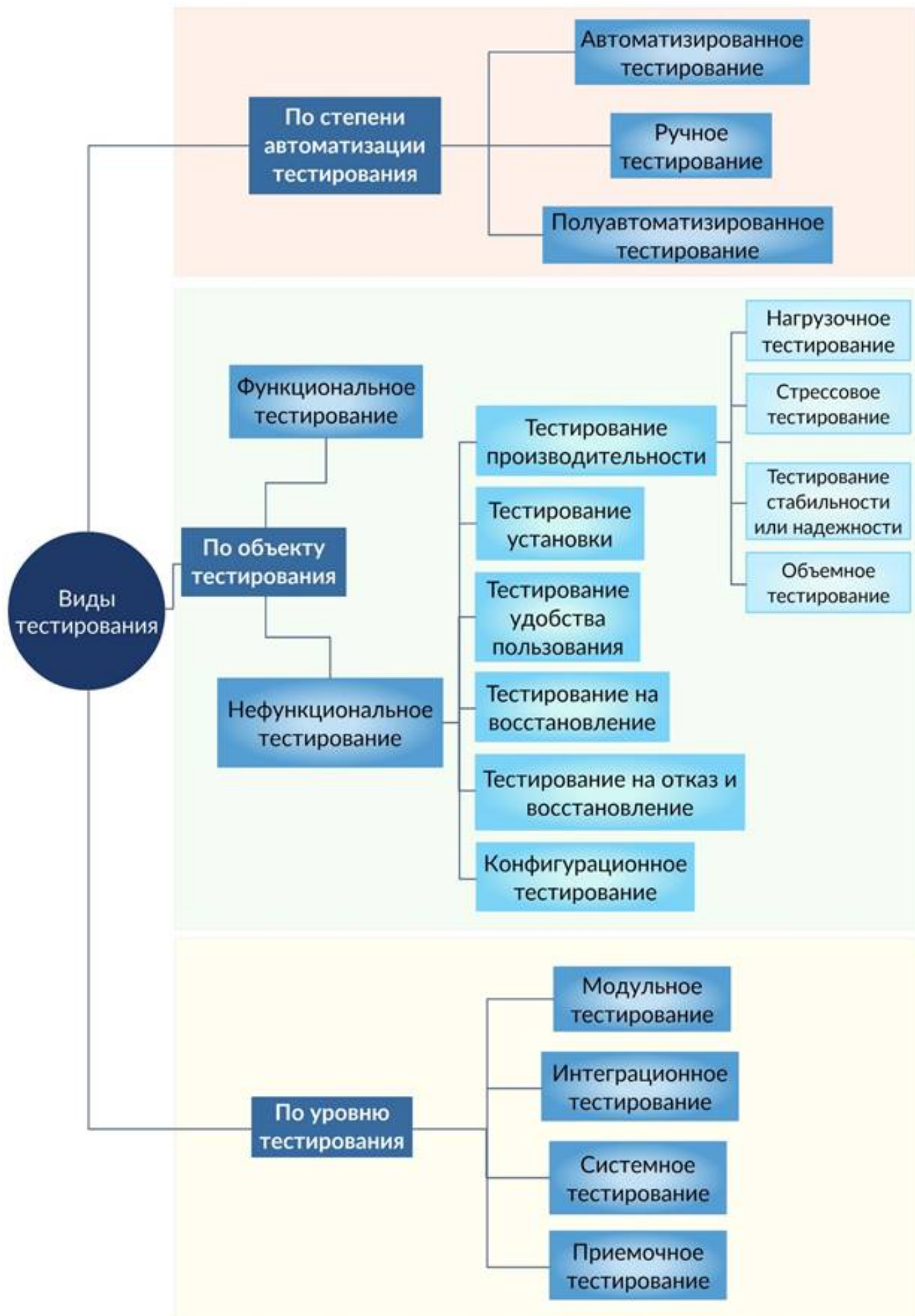


Рисунок 1. Виды тестирования ПО

По уровню тестирования выделяют модульное, интеграционное, системное и приемочное тестирование.

Модульное тестирование основано на проверке всех модулей системы по отдельности, т.е. для каждого модуля создается свой набор тестов и ставятся заглушки на другие модули, чтобы протестировать работоспособность конкретного модуля. Если тестирование всех модулей прошло успешно или все ошибки, обнаруженные в ходе данного тестирования, были исправлены, то все модули интегрируются в единую систему, которая в дальнейшем тоже будет тестироваться.

Интеграционное тестирование - это тестирование, которое проверяет архитектуру системы, полноту и корректность взаимодействия между модулями, компонентами системы или несколькими системами.

Системное тестирование анализирует поведение системы, выполнение ее функциональных и нефункциональных требований. Данное тестирование рассматривает работу пользовательских интерфейсов, совместимость системы с окружением, на сколько правильно выполняется функциональность системы и т.п., таким образом, основной задачей системного тестирования является обнаружение дефектов, которые влияют на работу системы в целом.

Приемочное тестирование - это комплексное проведение испытаний, которые подтверждают соответствие работы программного обеспечения с установленными ранее требованиями, что позволяет определить степень готовности продукта к эксплуатации. Благодаря приемочному тестированию можно обнаружить системные нарушения, а также проблемы, которые связаны с простотой и удобством использования ПО.

По объекту тестирования выделяют функциональное и нефункциональное тестирование.

Функциональное тестирование сопоставляет результат реализованных функций техническому заданию и требованиям, установленных на начальном этапе разработки продукта. Данный вид тестирования отслеживает как работу отдельных функций, так и их взаимодействие, а также корректность обработки входных и выходных параметров.

Нефункциональное тестирование - это тестирование, которое оценивает второстепенные качества функционирования программного обеспечения, например, такие как надежность, производительность, удобство, масштабируемость, безопасность и портируемость.

К нефункциональному тестированию относятся:

1) Тестирование производительности. Данное тестирование проводится для определения стабильности, доступности и надежности системы, а также для анализа поведения системы при максимальных и ожидаемых нагрузках. Тестирование производительности включает в себя такие подтипы как:

а) Нагрузочное тестирование. Вид тестирования, который имитирует пользовательскую нагрузку на систему, чтобы проанализировать правильность работы системы и влияние модулей системы на ее производительность, чтобы рассчитать стабильность и производительность при максимальных нагрузках;

б) Стрессовое тестирование. Позволяет оценить устойчивость и надежность системы в стрессовых условиях, т.е. в исключительных ситуациях, при которых происходят аварийные изменения, перегрузки, значительное повышение интенсивности выполнения задач. Также данное тестирование помогает проанализировать способность системы возвращаться в нормальное состояние;

в) Тестирование стабильности или надежности. Определяет способность системы

поддерживать оптимальную производительность с внушительными нагрузками при длительном использовании системы;

г) Объемное тестирование. Предназначено для анализа производительности системы при увеличении объема данных.

2) Тестирование установки. Данное тестирование позволяет соотнести степень готовности программного обеспечения к эксплуатации с точки зрения корректности ее установки в рабочую среду.

3) Тестирование удобства пользования. Это тестирование направлено на исследование уровня удобства, обучаемости и привлекательности в применении программного обеспечения.

4) Тестирование на отказ и восстановление. Анализирует надежность программного обеспечения за счет способности восстановления ПО после аварий, потраченного времени на это восстановление, а также обнаруживает возможные последствия после сбоев.

5) Конфигурационное тестирование. Вид тестирования, который проверяет работоспособность ПО под разными конфигурациями, т.е. на различных программных и аппаратных окружениях.

По степени автоматизации выделяют автоматизированное, ручное и полуавтоматизированное тестирование.

Автоматизированное тестирование. Выполнение тестов, реализуемое при помощи заранее записанной последовательности тестов [4, с. 9].

К плюсам автоматизированного тестирования относят экономию времени самого тестирования больших приложений, поскольку обработать значительный объем данных вручную гораздо сложнее, чем один раз написать сценарий выполнения тестирования. Из этого вытекает и второй плюс, уже написанный тестовый сценарий можно использовать в дальнейшем для проведения такого же рода тестирования, что в перспективе сэкономит время. Также большим преимуществом является то, что автоматизированное тестирование позволяет достаточно быстро смоделировать пользовательскую нагрузку на систему.

Но существуют и минусы. Во-первых, данное тестирование не способно охватить выполнение всех требований относительно пользовательского интерфейса. Во-вторых, существует ряд ошибок, которые может заметить только человек. И в-третьих, этот вид тестирования требует немалых затрат.

Ручное тестирование ПО представляет собой исполнение тестов без использования готовых программных тестовых наборов. Ручное тестирование является более гибким, т.к. существует возможность изменения параметров в программе во время проведения тестов.

При данном тестировании появляется возможность полностью протестировать пользовательский интерфейс, поскольку человек опираясь на накопленный опыт и знания способен обнаружить все нюансы, связанные с интерфейсом. Также реализуется подход исследовательского тестирования, т.е. возможности приложения проверяются спонтанно придуманными тестами. И конечно же, стоимость такого тестирования будет гораздо дешевле.

Минусами ручного тестирования является:

1) Нет возможности при повторной необходимости использовать тот же тест, который применялся ранее.

2) Нет возможности исполнения нагрузочного тестирования с большим числом пользователей.

3) Существует большая вероятность пропуска ошибки из-за невнимательности.

Полуавтоматизированное тестирование представляет собой сочетание ручного тестирования с автоматизированным. В этом подходе также есть преимущества и недостатки.

Большой плюс данного подхода заключается в возможности ручного тестирования для охвата всего пользовательского интерфейса, и программное использование для сокращения времени определенных случаев тестирования, повторения тестирования.

Минус полуавтоматизированного тестирования подразумевает неправильное применение метода, т.е. непонимание принципа сочетания ручного и автоматизированного тестирования, что может привести к более значительным временным и денежным затратам.

Таким образом, в виду многообразия тестов для проверки программного обеспечения, необходимо удостовериться в степени надобности определенных тестов и выбрать конкретный подход к тестированию. Как говорилось уже ранее, разные виды тестирования рассматриваются для разных целей, и рациональнее всего использовать не весь набор существующих тестов, а только такой набор, который обеспечил бы необходимое качество ПО и его соответствие выдвигаемым требованиям.

Список литературы:

1. Гленфорд Майерс. Искусство тестирования программ, 3-е издание. / Гленфорд Майерс, Том Баджетт, Кори Сандлер. – Москва: Диалектика, 2012. – 272 с.
2. Рекс Блэк. Ключевые процессы тестирования. Планирование, подготовка, проведение, совершенствование. / Рекс Блэк. – Москва: Лори, 2006. – 544 с.
3. Сэм Канер. Тестирование программного обеспечения. Фундаментальные концепции менеджмента бизнес-приложений: Пер. с англ. / Сэм Канер, Джек Фолк, Енг Кек Нгуен. – Москва: ДиаСофт, 2001. – 544 с.
4. Эрик Ван Венендал. Стандартный глоссарий терминов, используемых в тестировании программного обеспечения, версия 2.3 // International Software Testing Qualifications Board (ISTQB), 2014.