Принципы и методы тестирования ПО

Классификация **по целям**:

* Функциональное тестирование (functional testing) рассматривает заранее указанное поведение и основывается на анализе спецификации компонента или системы в целом.
* Нефункциональное тестирование (non-functional testing) — тестирование атрибутов компонента или системы, не относящихся к функциональности.
* Тестирование пользовательского интерфейса (GUI Testing) — проверка интерфейса на соответствие требованиям.
* Тестирование удобства использования (Usability Testing) — определение степени удобства использования, обучаемости, понятности и привлекательности для пользователей разрабатываемого продукта в контексте заданных условий.
* Тестирование безопасности (security testing) — проверка безопасности системы, а также анализ рисков, связанных с обеспечением целостного подхода к защите приложения, атак хакеров, вирусов, несанкционированного доступа к конфиденциальным данным.
* Инсталляционное тестирование (installation testing) – проверка успешной установки и настройки, а также обновления или удаления приложения.
* Конфигурационное тестирование (Configuration Testing) — проверка работы ПО при различных конфигурациях системы.
* Тестирование на отказ и восстановление (Failover and Recovery Testing) – проверка обеспечения сохранности и целостности данных, после возможных сбоев, возникших в связи с ошибками ПО, отказами оборудования или проблемами связи.
* Тестирование локализации (localization testing) — проверка адаптации ПО для определенной аудитории в соответствии с ее культурными особенностями.
* Тестирование производительности (performance testing) — определение стабильности и потребления ресурсов в условиях различных сценариев использования и нагрузок.
* Нагрузочное тестирование (load testing) — определение или сбор показателей производительности и времени отклика программно-технической системы или устройства в ответ на внешний запрос с целью установления соответствия требованиям, предъявляемым к данной системе.
* Тестирование стабильности или надежности (Stability / Reliability Testing) — проверка работоспособности приложения при длительном (многочасовом) тестировании со средним уровнем нагрузки.
* Стрессовое тестирование (Stress Testing) – проверка приложения и системы на работоспособность в условиях стресса (например, повышение интенсивности выполнения операций до очень высоких значений или аварийное изменение конфигурации сервера) и также оценка способности системы к регенерации, т.е. к возвращению к нормальному состоянию после прекращения воздействия стресса.
* Объемное тестирование (Volume Testing) — тестирование, которое проводится для получения оценки производительности при увеличении объемов данных в базе данных приложения.
* Тестирование масштабируемости (scalability testing) — тестирование, которое измеряет производительность сети или системы, когда количество пользовательских запросов увеличивается или уменьшается.

Классификация **по уровню тестирования**:

* Модульное (компонентное) тестирование (Unit Testing) проводится самими разработчиками, т.к. предполагает полный доступ к коду, для тестирования какого-либо одного логически выделенного и изолированного элемента (модуля) системы в коде, проверяет функциональность и ищет дефекты в частях приложения, которые доступны и могут быть протестированы по-отдельности (модули программ, объекты, классы, функции и т.д.).
* Интеграционное тестирование (Integration Testing) направлено на проверку корректности взаимодействия нескольких модулей, объединенных в единое целое, т.е. проверяется взаимодействие между компонентами системы после проведения компонентного тестирования.
* Системное тестирование (System Testing) — это проверка как функциональных, так и не функциональных требований в системе в целом. При этом выявляются дефекты и оцениваются характеристики качества системы — ее устойчивость, надежность, безопасность и производительность.
* Операционное тестирование (Release Testing). Даже если система удовлетворяет всем требованиям, важно убедиться в том, что она удовлетворяет нуждам пользователя и выполняет свою роль в среде своей эксплуатации. Поэтому так важно провести операционное тестирование как финальный шаг валидации. Кроме этого, тестирование в среде эксплуатации позволяет выявить и нефункциональные проблемы, такие как: конфликт с другими системами, смежными в области бизнеса или в программных и электронных окружениях и др. Очевидно, что нахождение подобных вещей на стадии внедрения — критичная и дорогостоящая проблема.

Классификация **по исполнению кода**:

* Статическое тестирование — процесс тестирования, который проводится для верификации практически любого артефакта разработки. Целью анализа является раннее выявление ошибок и потенциальных проблем в продукте.
* Динамическое тестирование проводится на работающей системе.

Классификация **по хронологии выполнения**:

* Повторное/подтверждающее тестирование (re-testing/confirmation testing) — тестирование, во время которого исполняются тестовые сценарии, выявившие ошибки во время последнего запуска, для подтверждения успешности исправления этих ошибок.
* Регрессионное тестирование (regression testing) — это тестирование после внесения изменений в код приложения, для подтверждения того факта, что эти изменения не внесли ошибки в областях, которые не подверглись изменениям.
* Приёмочное тестирование проверяет соответствие системы потребностям, требованиям и бизнес-процессам пользователя.

Традиционно тестирование программного обеспечения можно разделить на различные категории по разным параметрам, которые были рассмотрены выше. Однако наиболее популярной классификацией является разделение на ручное и автоматизированное тестирование.

Ручное тестирование — это вид тестирования программного обеспечения, при котором тесты выполняются тестировщиком вручную, без использования каких-либо средств автоматизации и полностью зависит от тестировщика.

Когда используется ручное тестирование:

1. продукт находится на начальном этапе разработки;
2. продукт краткосрочный и небольшой;
3. при тестировании удобства использования продукта;
4. при интуитивном или исследовательском тестировании;
5. при работе с физическими продуктами.

Автоматизированное тестирование — это метод тестирования программного обеспечения, который предполагает использование инструментов и фреймворков автоматизации для выполнения одного и того же набора тест-кейсов снова и снова.

Когда используется автоматизированное тестирование:

1. при выполнении повторяющихся тестов;
2. при использовании тестирования производительности или при нагрузочном тестировании;
3. когда имеется большое количество тест-кейсов;
4. когда необходимо исключить человеческий фактор;
5. при работе с большими объемами данных.

Источник: Хабр / [Электронный ресурс] // Теория тестирования ПО просто и понятно : [сайт]. — URL: https://habr.com/ru/articles/587620/ (дата обращения: 06.11.2023).

<https://habr.com/ru/articles/716402/>