Тестирование

Тестирование (testing) является методом верификации, в рамках которого результаты работы тестируемой системы или компонента в ситуациях из выделенного конечного набора проверяются на соответствие проектным решениям, требованиям, общим задачам проекта, в рамках которого эта система разрабатывается или сопровождается. Ситуации, в которых выполняется тестирование, называют тестовыми ситуациями (test situations, test purposes), а процедуры, описывающие процесс создания этих ситуаций и проверки, которые необходимо выполнить над полученными результатами, — тестами.

Часто говорят, что тестирование должно проверять соответствие работы ПО требованиям к ней, однако «требования» здесь нужно понимать очень широко, в частности, включать в них детали проектных решений, определяемых на достаточно поздних этапах проекта. Довольно часто проводят тестирование, в котором проверяют только то, что система не «падает», не создает исключительных ситуаций или «странных», совершенно неожиданных для разработчиков результатов. Все эти ограничения обычно можно считать несформулированной явно, но подразумеваемой частью требований (стоит помнить, однако, что системы некоторых типов в определенных ситуациях должны «падать» или создавать исключительные ситуации). Кроме того, чем меньше при тестировании задействованы достаточно точные требования к тому, что система должна делать, чем более оно ориентировано лишь на поиск сбоев или других видов инцидентов, тем меньше пользы оно приносит проекту. Поэтому такое тестирование оправдано только как прелюдия к серьезному тестированию, для проверки того, что система в целом работоспособна, или в тех ситуациях, где описание требований вообще отсутствует, или же они сформулированы крайне нечетко и неполно.

Тестирование может использоваться и для валидации, в этом случае оно проверяет соответствие поведения ПО ожиданиям и потребностям пользователей и заказчиков, а для оценки этого соответствия должны привлекаться сами пользователи, их представители, бизнес-аналитики или эксперты в предметной области.

Тестирование, как и верификация вообще, служит для поиска ошибок или дефектов и для оценки качества ПО. Эффективность решения обеих этих задач во многом определяется тем, какой именно набор тестовых ситуаций выбран для проведения тестирования. Чтобы иметь некоторые гарантии аккуратности полученных в ходе тестирования оценок качества, необходимо выбирать тестовые ситуации систематическим образом, в соответствии с основными задачами и рисками проекта. Правила, определяющие набор необходимых тестовых ситуаций, называют критериями полноты (или адекватности) тестирования (test adequacy criteria). Обычно такой критерий использует разбиение всех возможных при работе проверяемого ПО ситуаций на некоторые классы эквивалентности, такие, что ситуации из одного класса достаточно похожи друг на друга и работа ПО в них не должна отличаться сколь-нибудь значительным образом. Полноту тестирования при этом можно определять по проценту задействованных, «покрытых» в нем классов ситуаций. Такой критерий полноты называется критерием тестового покрытия (test coverage criterion), а процент покрытых в результате работы тестов классов ситуаций — достигнутым тестовым покрытием.

Заметим, что различных критериев полноты довольно много (см. ниже), а тестирование, покрывшее 100% выделенных по одному из таких критериев классов ситуаций, и поэтому часто называемое полным, в действительности не является полным или исчерпывающим в каком-либо смысле, связанным со строгими гарантиями отсутствия ошибок и соответствия ПО требованиям. Такого рода исчерпывающее тестирование невозможно для практически значимых систем.

Подготовка и проведение тестирования в проекте создания или сопровождения ПО проходят примерно по следующему плану.

* Определение целей тестирования (test objectives, test goals) на основе задач и рисков проекта. Такие цели очерчивают проверяемые характеристики и свойства ПО, тщательность тестирования отдельных компонентов и подсистем. Они также определяют используемые в проекте виды тестирования и методы построения тестов.
* Определение требований, свойств и ограничений, которые должны проверяться в ходе тестирования.
* Определение критерия полноты тестирования, который будет использоваться в данном проекте. Критерий полноты должен быть согласован с целями тестирования, он управляет выбором тестовых ситуаций для тестирования, а также определяет, когда можно прекратить тестирование.
* Построение набора тестов, нацеленных на достижение выбранного критерия полноты и проверяющих определенные ранее требования и ограничения.
* Отладка, выполнение тестов и получение итоговых результатов тестирования в виде сообщений о выполненных действиях и нарушениях проверяемых ограничений. Отладка тестов включает пробные прогоны, устранение обнаруживаемых ошибок в самих тестах, а также описаниях проектных решений и требований, на основе которых получены тесты. По результатам отладки может потребоваться пополнение набора тестов для достижения максимального возможной полноты тестирования в рамках выделенных ресурсов.
* Анализ результатов тестирования, составление отчетов о найденных дефектах и о полученных оценках качества ПО.

Виды тестирования

Классификация видов тестирования достаточно сложна, потому что может проводиться по нескольким разным аспектам.

* По уровню или масштабу проверяемых элементов системы тестирование делится на следующие виды.
  + Модульное или компонентное (unit testing, component testing) — проверка корректности работы отдельных компонентов системы, выполнения ими своих функций и предполагаемых проектом характеристик.
  + Интеграционное (integration testing) — проверка корректности взаимодействий внутри отдельных групп компонентов.
  + Системное (system testing) — проверка работы системы в целом, выполнения ею своих основных функций, с использованием определенных ресурсов, в окружении с заданными характеристиками.
* По проверяемым характеристикам качества тестирование может быть тестирование функциональности, производительности (и по времени, и по другим ресурсам), надежности, переносимости или удобства использования. Более специфические виды тестирования, нацеленные на оценку отдельных атрибутов, — тестирование защищенности, совместимости или восстановления при сбоях. Специфическим видом тестирования, нацеленным на минимизацию риска того, что в результате доработки или внесения ошибок качество системы изменилось в худшую сторону, является регрессионное тестирование (regression testing). При его проведении используется уже применявшийся ранее набор тестов, и оно должно выявить различия между результатами, полученными на этих тестах ранее, и наблюдаемыми после внесения изменений.
* По источникам данных, используемых для построения тестов, тестирование относится к одному из следующих видов.
  + Тестирование черного ящика (black-box testing, часто также называется тестированием соответствия, conformance testing, или функциональным тестированием, functional testing) — нацелено на проверку соблюдения требований. Использует критерии полноты, основанные на требованиях, и техники построения тестов, использующие только информацию, заданную в требованиях к проверяемой системе. Частными случаями этого вида тестирования являются тестирование на соответствие стандартам и квалификационное или сертификационное тестирование, нацеленное на получение некоторого сертификата соответствия определенным требованиям или стандартам.
  + Тестирование белого ящика (white-box testing, glass-box testing, также структурное тестирование, structural testing)— нацелено на проверку корректности работы кода. Использует критерии полноты и техники построения тестов, основанные на структуре проверяемой системы, ее исходного кода.
  + Тестирование серого ящика (grey-box testing) использует для построения тестов как информацию о требованиях, так и коде. На практике оно встречается чаще, чем предыдущие крайние случаи.
  + Тестирование, нацеленное на ошибки — использует для построения тестов гипотезы о возможных или типичных ошибках в ПО такого же типа, как проверяемое. К этому типу относятся, например, следующие виды тестирования.
    - Тестирование работоспособности (sanity testing, smoke testing), нацеленное на проверку того, в систему включены все ее компоненты и операции, и система не дает сбоев при выполнении своих основных функций в простейших сценариях использования.
    - Тестирование на отказ, пытающееся найти ошибки в ПО, связанные с контролем корректности входных данных.
    - Нагрузочное тестирование (load testing), проверяющее работоспособность ПО при больших нагрузках — больших объемах входных, выходных или промежуточных данных, большой сложности решаемых задач, большом количестве пользователей, работающих с ПО и пр.
    - Тестирование в предельных режимах (stress testing), проверяющее работоспособность ПО на границах его возможностей и на границах той области, где оно должно использоваться.
* По роли команды, выполняющей тестирование, оно может относиться к следующим видам.
  + Внутреннее тестирование выполняется в рамках проекта по разработке системы силами организации-разработчика ПО.
  + Независимое тестирование выполняется третьими лицами (не разработчиками, не заказчиками и не пользователями) для получения объективных и аккуратных оценок качества системы.
  + Аттестационное тестирование (приемочные испытания) выполняется представителями заказчика непосредственно перед приемкой системы в эксплуатацию для проверки того, что основные функции системы реализованы. Обычно аттестационные тесты являются очень простыми тестами функциональности и производительности.
  + Пользовательское тестирование осуществляется силами пользователей системы. У него есть два часто упоминаемых частных случая.
    - Альфа-тестирование (alpha-testing) выполняется самими разработчиками, но в среде, максимально приближенной к рабочему окружению системы и на наиболее вероятных сценариях ее реального использования.
    - Бета-тестирование (beta-testing) выполняется пользователями, желающими познакомиться с возможностями системы до ее официального выпуска и передачи в эксплуатацию.