

Статические и динамические методы тестирования

Статическое тестирование является методикой тестирования программного обеспечения, при которой ПО тестируется без запуска кода. Оно представляет собой процесс или технику, которые выполняются для поиска потенциальных дефектов в программном обеспечении. Это также процесс обнаружения и устранения ошибок и дефектов в различных сопроводительных документах, таких как спецификации требований к программному обеспечению и т. д.

Можно поделить статическое тестирование на 2 типа:

1. **Обзоры (Review);**
2. **Статический анализ (Static Analysis).**

Обзоры (Review) – проверка обычно используется для поиска и устранения ошибок или неясностей в документах. Это могут быть требования, дизайн, тестовые случаи и так далее.

В свою очередь обзоры делятся на:

- **Неформальные.** При неофициальном рассмотрении создатель документов показывает содержание документов аудитории. Каждый присутствующий высказывает свое мнение, что позволяет выявить недостатки на ранней стадии.
- **Сквозные просмотры (Walkthroughs).** Выполняются опытным человеком или экспертом для проверки отсутствия дефектов, с целью предупреждения возникновения проблем на этапе разработки или тестирования.
- **Экспертная оценка.** Означает проверку документов для выявления и исправления дефектов. В основном это делается в команде.
- **Инспектирование ПО.** Это, в большинстве случаев, проверка документа вышестоящим органом, например, проверка требований к программному обеспечению.

Статический анализ (Static Analysis) – код, написанный разработчиками, анализируется на наличие структурных дефектов, которые могут привести к ошибкам.

Статический анализ включает оценку качества кода, написанного разработчиками. Для анализа кода и сравнения его со стандартом используются разные инструменты. Статический анализ хорошо помогает найти такие ошибки, как:

- неиспользуемые переменные;
- мертвый код;
- бесконечные циклы;
- переменные с неопределенными значениями;

- неправильный синтаксис.

Статический анализ состоит из 3-х частей:

1. Поток данных (поток данных связан с обработкой потока);
2. Контроль потока (как выполняются операторы или инструкции);
3. Цикломатическая сложность (измерение сложности программы, которое в основном связано с количеством независимых путей в графе потоков управления программы).

Анализ может производиться как вручную, так и с помощью специальных инструментов.

Целью анализа является наиболее раннее выявление ошибок и потенциальных проблем в программном продукте. Как правило, код ревью выполняется самим разработчиком.

Преимущества статического тестирования

- Снижает стоимость фикса найденных багов, поскольку выявляет баги на ранних этапах цикла разработки программного обеспечения.
- Отзывы, полученные в ходе этого тестирования, помогают улучшить функционирование процесса, что также помогает команде избежать подобных дефектов и багов.
- Повышает информированность о различных проблемах качества программного обеспечения.
- Улучшает обмен критической и важной информацией между членами команды.
- Существенно сокращаются усилия по исправлению ошибок, что еще больше способствует продуктивности разработки.

Недостатки статического тестирования

- Процесс статического тестирования может занимать много времени, так как в основном он выполняется вручную.
- Препятствует обнаружению уязвимостей, представленных в среде выполнения.

Метод динамического тестирования – тип тестирования, который проверяет функциональность приложения, когда код выполняется. Проще говоря, динамическое тестирование выполняется путем фактического использования приложения и определения того, работает ли функциональность так, как ожидается.

Динамическое тестирование включает в себя тестирование ПО в режиме реального времени путем предоставления входных данных и изучения результата поведения программы.

Можно привести такой простой пример – функционал для входа в систему. При создании учетной записи и пароля к ней будут присутствовать

определенные правила для создания надежного пароля. Например, пароль должен быть не менее 6 символов, иметь заглавную букву и хотя бы один специальный символ. Это не что иное, как различные условия или параметры, и если пользователь вводит любое значение, которое отличается от этих условий, приложение должно его отклонить. При тестировании данной функциональности нужно ввести все необходимые условия, а затем проверить результат.

Если рассмотреть функции, предлагаемые динамическим тестированием, можно легко понять причины его выполнения в течение жизненного цикла тестирования программного обеспечения. С помощью этого тестирования команда может проверить различные критические аспекты программного обеспечения. Если оставить их без какой-либо оценки, они могут повлиять на производительность, функциональность, а также надежность программного продукта.

Преимущества динамического тестирования

- Это тщательное исследование, которое рассматривает всю функциональность приложения, поэтому качество соответствует самым высоким стандартам.
- Процесс динамического тестирования хорошо налажен, приложение тестируется с точки зрения пользователя, что повышает качество ПО.
- Обнаружение сложных ошибок, которые могли ускользнуть на этапе код ревью.
- Динамическое тестирование может быть автоматизировано с помощью специальных инструментов.

Недостатки динамического тестирования

- Поскольку динамическое тестирование представляет собой сложный процесс, оно занимает много времени.
- Высокая стоимость проведения тестирования.
- Динамическое тестирование обычно выполняется после завершения кодирования, и найденные баги обнаруживаются позже в жизненном цикле разработки.

Таблица сравнения динамического и статического тестирования

Динамическое тестирование	Статическое тестирование
Этап валидации ПО	Этап верификации ПО
Включает выполнение программного кода	Не требует выполнения программного кода
Обеспечивает функциональность продукта	Ориентировано на предотвращение дефектов
Выполняется на более поздних этапах разработки программного обеспечения	Выполняется на ранних этапах разработки программного обеспечения
Высокая стоимость исправления багов	Более низкая стоимость исправления багов
Покрывает ограниченную область кода, требует меньшего охвата	Обеспечивает более широкий охват, чем динамическое тестирование, за более короткий промежуток времени
Включает как функциональное, так и нефункциональное тестирование	Включает в себя различные методы оценки, сквозной просмотр, ревью и многое другое
Цель – поиск и устранение багов	Цель – предотвращение дефектов программного обеспечения
Обнаруживается меньше дефектов, чем при статическом тестировании	Комплексное тестирование кода, которое помогает найти больше дефектов в системе
Выполняется после деплоя кода	Выполняется перед деплоем кода

Исходя из вышеизложенного, можно легко сделать вывод – несмотря на то, что статическое тестирование требует много времени на бурные дискуссии и встречи, все же стоит потратить время на предотвращение появления дефектов на последних этапах разработки продукта. Поэтому статическое тестирование по праву считается важным шагом на пути к разработке ПО без ошибок. Но важность динамического тестирования также огромна. Благодаря

непосредственному выполнению тестов программного обеспечения (проверки функционального поведения, производительности, надежности и других важных аспектов) команда может проверить и подтвердить качество и эффективность ПО.