

Лекция 2. Виды и уровни тестирования (модульное, интеграционное, системное)

Программная система строится иерархически: из отдельных функций и классов собираются модули, из модулей — подсистемы, из подсистем — целостный продукт, который затем передаётся заказчику. Каждому уровню этой иерархии соответствует свой уровень тестирования со своими целями, объектами проверки, ответственными ролями и типичными видами дефектов. Понимание уровней и видов тестирования позволяет выстроить стратегию проверки качества, в которой каждый дефект обнаруживается на том уровне, где его поиск и исправление обходятся дешевле всего.

1. Модульное (компонентное) тестирование

Модульное тестирование (unit testing) проверяет наименьшие тестируемые части программы — функции, методы, классы — изолированно от остальной системы. Изоляция достигается с помощью тестовых двойников: заглушек (stubs), имитаций (mocks) и подделок (fakes), которые заменяют реальные зависимости — базы данных, сетевые сервисы, файловую систему.

Модульные тесты обычно пишут сами разработчики с использованием фреймворков xUnit-семейства: JUnit для Java, pytest для Python, NUnit для C#. Ключевые достоинства уровня — высокая скорость выполнения (тысячи тестов за секунды), точная локализация дефекта и возможность запуска при каждом изменении кода. Типичные объекты проверки: граничные условия, обработка некорректных входных данных, ветвления логики, исключительные ситуации.

2. Интеграционное тестирование

Интеграционное тестирование проверяет взаимодействие между компонентами и системами. Даже если каждый модуль по отдельности работает корректно, на стыках возникают специфические дефекты: несогласованность форматов данных, неверные контракты интерфейсов, ошибки обработки таймаутов, проблемы конкурентного доступа.

Различают несколько стратегий интеграции:

- «Большой взрыв» (big bang) — все компоненты объединяются одновременно; просто организовать, но сложно локализовать дефекты;
- Нисходящая (top-down) — интеграция начинается с верхних уровней, нижние заменяются заглушками;
- Восходящая (bottom-up) — сначала объединяются низкоуровневые модули, вызовы сверху имитируются драйверами;
- Инкрементная/непрерывная — компоненты подключаются по одному по мере готовности, что характерно для современных CI/CD-процессов.

Отдельно выделяют системную интеграцию — проверку взаимодействия продукта с внешними системами: платёжными шлюзами, государственными сервисами, сторонними API.

3. Системное и приёмочное тестирование

Системное тестирование рассматривает продукт как единое целое в окружении, максимально приближенном к промышленному. Проверяются сквозные (end-to-end) сценарии поведения системы, а также нефункциональные характеристики: производительность, надёжность, безопасность, удобство использования. Ответственность за этот уровень обычно несёт независимая команда тестирования.

Приёмочное тестирование (acceptance testing) отвечает на вопрос о готовности системы к внедрению и проводится с участием заказчика и пользователей. Его разновидности: пользовательское приёмочное тестирование (UAT), эксплуатационное приёмочное тестирование, а также альфа- и бета-тестирование, когда продукт проверяется соответственно внутри организации и ограниченной группой внешних пользователей.

4. Виды тестирования: классификационные оси

Помимо уровней, тестирование классифицируют по ряду независимых осей. По целям выделяют функциональное тестирование (что делает система) и нефункциональное (насколько хорошо она это делает: нагрузка, безопасность, юзабилити). По знанию внутренней структуры — методы «чёрного ящика» (только внешнее поведение), «белого ящика» (анализ кода и покрытия) и «серого ящика» (частичное знание архитектуры). По степени автоматизации — ручное и автоматизированное тестирование. По отношению к изменениям — подтверждающее тестирование (re-testing, проверка конкретного исправления) и регрессионное (проверка того, что изменения не сломали существующую функциональность). Эти оси независимы: например, регрессионное тестирование может быть функциональным, автоматизированным и проводиться на системном уровне одновременно.

Выводы

Уровни тестирования образуют последовательность рубежей контроля качества: модульный уровень ловит локальные ошибки логики, интеграционный — дефекты взаимодействия, системный — проблемы поведения продукта в целом, приёмочный — расхождения с ожиданиями заказчика. Грамотная стратегия распределяет усилия между уровнями так, чтобы каждый дефект обнаруживался как можно раньше и дешевле.

Контрольные вопросы

7. Что является объектом проверки на модульном уровне тестирования? Какие инструменты для этого применяются?

8. Зачем нужны тестовые двойники (mocks, stubs) и чем они различаются?

9. Сравните стратегии интеграции «большой взрыв», top-down и bottom-up.

10. Чем системное тестирование отличается от приёмочного?

11. Что такое альфа- и бета-тестирование?

12. В чём разница между подтверждающим и регрессионным тестированием?