

Примеры модульного тестирования

Уровни процесса верификации. Задачи и цели модульного тестирования. Понятие модуля и его границ. Тестирование классов. Подходы к проектированию тестового окружения. Организация модульного тестирования с Unit Test на платформе .Net.

Процесс верификации активен в течение практически всего жизненного цикла системы и работает параллельно с процессом разработки. Разработка системы, как правило, идет на различных уровнях: вначале разрабатывается концепция системы, системные требования, затем архитектура системы, ее разбиение на модули, затем разрабатываются отдельные модули. Последовательность этих уровней зависит от типа жизненного цикла, но их состав практически всегда одинаков. Процесс верификации также разбивается на отдельные уровни:

- системное тестирование, в ходе которого тестируется система в целом;
- интеграционное тестирование, в ходе которого тестируются группы взаимодействующих модулей и компонент системы;
- модульное тестирование, в ходе которого тестируются отдельные компоненты.

Цели и задачи модульного тестирования

Каждая сложная программная система состоит из отдельных частей - модулей, выполняющих ту или иную функцию в составе системы.

Для каждого модуля, подвергаемого тестированию, разрабатывается тестовое окружение, включающее в себя драйвер и заглушки, готовятся тест-требования и тест-планы, описывающие конкретные тестовые примеры.

Основная цель модульного тестирования - удостовериться в соответствии требованиям каждого отдельного модуля системы перед тем, как будет произведена его интеграция в состав системы.

При этом в ходе модульного тестирования решаются следующие основные задачи [Белов С Модульное тестирование и Test-Driven Development, или Как управлять страхом в программировании IT News, #21/2005]:

1. Поиск и документирование несоответствий требованиям

2. Поддержка разработки и рефакторинга низкоуровневой архитектуры системы и межмодульного взаимодействия
3. Поддержка рефакторинга модулей
4. Поддержка устранения дефектов и отладки

Первая задача - классическая задача тестирования, включающая в себя не только разработку тестового окружения и тестовых примеров, но и выполнение тестов, протоколирование результатов выполнения, составление отчетов о проблемах.

Вторая задача больше свойственна "легким" методологиям типа XP, где применяется принцип тестирования перед разработкой (Test-driven development), при котором основным источником требований для программного модуля является тест, написанный до реализации самого модуля. Однако, даже при классической схеме тестирования модульные тесты могут выявить проблемы в дизайне системы и нелогичные или запутанные механизмы работы с модулем.

Третья задача связана с поддержкой процесса изменения системы. Достаточно часто в ходе разработки требуется проводить рефакторинг модулей или их групп - оптимизацию или полную переделку программного кода с целью повышения его сопровождаемости, скорости работы или надежности. Модульные тесты при этом являются мощным инструментом для проверки того, что новый вариант программного кода выполняет те же функции, что и старый.

Последняя, четвертая, задача сопряжена с обратной связью, которую получают разработчики от тестировщиков в виде отчетов о проблемах. Подробные отчеты о проблемах, составленные на этапе модульного тестирования, позволяют локализовать и устранить многие дефекты в программной системе на ранних стадиях ее разработки или разработки ее новой функциональности.

В силу того, что модули, подвергаемые тестированию, обычно невелики по размеру, модульное тестирование считается наиболее простым (хотя и достаточно трудоемким) этапом тестирования системы. Однако, несмотря на внешнюю простоту, с модульным тестированием сопряжены две проблемы.

Первая из них связана с тем, что не существует единых принципов определения того, что в точности является отдельным модулем.

Вторая заключается в различиях в трактовке самого понятия модульного тестирования - понимается ли под ним обособленное тестирование модуля, работа которого поддерживается только тестовым окружением, или речь идет о проверке корректности работы модуля в составе уже разработанной системы. В последнее время термин "модульное тестирование" чаще используется во втором смысле, хотя в этом случае речь скорее идет об интеграционном тестировании.