МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«Гжельский государственный университет»** (ГГУ)

Колледж ГГУ

Специальность 09.02.07 Информационные системы и программирования

**Реферат**

**По дисциплине «технология разработки программного обеспечения»**

**на тему «Тестовое покрытие»**

ВЫПОЛНИЛ:

Студент группы ИСП-О-17

Шашков И.С.

ПРОВЕРИЛА:

Прокуронова А.Ю.

Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

п. Электроизолятор

1. г.

**Тестовое Покрытие (Test Coverage)**

**Тестовое Покрытие** - это одна из метрик оценки качества тестирования, представляющая из себя плотность покрытия тестами требований либо исполняемого кода.

Если рассматривать тестирование как "проверку соответствия между реальным и ожидаемым поведением программы, осуществляемая на конечном наборе тестов", то именно этот конечный набор тестов и будет определять тестовое покрытие:

*Чем выше требуемый уровень тестового покрытия, тем больше тестов будет выбрано, для проверки тестируемых требований или исполняемого кода.*

Сложность современного программного обеспечения и инфраструктуры сделало невыполнимой задачу проведения тестирования со 100% тестовым покрытием. Поэтому для разработки набора тестов, обеспечивающего более менее высокий уровень покрытия можно использовать специальные инструменты либо техники тест дизайна.

Существуют следущие подходы к оценке и измерению тестового покрытия:

1. [**Покрытие требований (Requirements Coverage)**](http://www.protesting.ru/testing/testcoverage.html#requirements) - оценка покрытия тестами функциональных и нефункциональных требований к продукту путем построения матриц трассировки (traceability matrix).
2. [**Покрытие кода (Code Coverage)**](http://www.protesting.ru/testing/testcoverage.html#code) - оценка покрытия исполняемого кода тестами, путем отслеживания непроверенных в процессе тестирования частей программного обеспечения.
3. [**Тестовое покрытие на базе анализа потока управления**](http://www.protesting.ru/testing/testcoverage.html#flow) - оценка покрытия основанная на определении путей выполнения кода программного модуля и создания выполняемых тест кейсов для покрытия этих путей.

**Различия**:  
Метод покрытия требований сосредоточен на проверке соответствия набора проводимых тестов требованиям к продукту, в то время как анализ покрытия кода - на полноте проверки тестами, разработанной части продукта (исходного кода), а анализ потока управления - на прохождении путей в графе или модели выполнения тестируемых функций (Control Flow Graph).

**Ограничения**:  
Метод оценки покрытия кода не выявит нереализованные требования, так как работает не с конечным продуктом, а с существующим исходным кодом.  
Метод покрытия требований может оставить непроверенными некоторые участки кода, потому что не учитывает конечную реализацию.

**Покрытие требований (Requirements Coverage)**

Расчет тестового покрытия относительно требований проводится по формуле:

Tcov = (Lcov/Ltotal) \* 100%

где:  
**Tcov** - тестовое покрытие  
**Lcov** - количество требований, проверяемых тест кейсами  
**Ltotal** - общее количество требований

Для измерения покрытия требований, необходимо проанализировать требования к продукту и разбить их на пункты. Опционально каждый пункт связывается с тест кейсами, проверяющими его. Совокупность этих связей - и является матрицей трассировки. Проследив связи, можно понять какие именно требования проверяет тестовый случай.

Тесты не связанные с требованиями не имеют смысла. Требования, не связанные с тестами - это "белые пятна", т.е. выполнив все созданные тест кейсы, нельзя дать ответ реализовано данное требование в продукте или нет.

Для оптимизации тестового покрытия при тестировании на основании требований, наилучшим способом будет использование стандартных техник тест дизайна. Пример разработки тестовых случаев по имеющимся требованиям рассмотрен в разделе: "[**Практическое применение техник тест дизайна при разработке тест кейсов**](http://www.protesting.ru/testing/testdesign_practice.html)"

**Покрытие кода (Code Coverage)**

Расчет тестового покрытия относительно исполняемого кода программного обеспечения проводится по формуле:

Tcov = (Ltc/Lcode) \* 100%

где:  
**Tcov** - тестовое покрытие  
**Ltc** - кол-ва строк кода, покрытых тестами  
**Lcode** - общее кол-во строк кода.

В настоящее время существует инструментарий (например: [**Clover**](http://www.atlassian.com/software/clover/)), позволяющий проанализировать в какие строки были вхождения во время проведения тестирования, благодаря чему можно значительно увеличить покрытие, добавив новые тесты для конкретных случаев, а также избавиться от дублирующих тестов. Проведение такого анализа кода и последующая оптимизация покрытия достаточно легко реализуется в рамках тестирования белого ящика (white-box testing) при модульном, интеграционном и системном тестировании; при тестировании же черного ящика (black-box testing) задача становится довольно дорогостоящей, так как требует много времени и ресурсов на установку, конфигурацию и анализ результатов работы, как со стороны тестировщиков, так и разработчиков.