



**Carnegie Mellon
Software Engineering Institute**

**Arcade Game Maker
Pedagogical Product Line:
Unit Test Plan Template**

Arcade Game Team

July 2003

This work is sponsored by the U.S. Department of Defense.

The Software Engineering Institute is a federally funded research and development center sponsored by the U.S. Department of Defense.

Copyright 2005 Carnegie Mellon University.

NO WARRANTY

THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN "AS-IS" BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

Use of any trademarks in this report is not intended in any way to infringe on the rights of the trademark holder.

Internal use. Permission to reproduce this document and to prepare derivative works from this document for internal use is granted, provided the copyright and "No Warranty" statements are included with all reproductions and derivative works.

External use. Requests for permission to reproduce this document or prepare derivative works of this document for external and commercial use should be addressed to the SEI Licensing Agent.

This work was created in the performance of Federal Government Contract Number F19628-00-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center. The Government of the United States has a royalty-free government-purpose license to use, duplicate, or disclose the work, in whole or in part and in any manner, and to have or permit others to do so, for government purposes pursuant to the copyright license under the clause at 252.227-7013.

For information about purchasing paper copies of SEI reports, please visit the publications portion of our Web site (<http://www.sei.cmu.edu/publications/pubweb.html>).

Contents

1	Overview	1
1.1	Identification	1
1.2	Document Map	1
1.3	Concepts	2
1.4	Readership	2
2	Template for the Test Plan	3
2.1	Test Suspension Criteria and Resumption Requirements	3
2.2	Problem Determination and Correction Responsibilities	4
3	Analyses and Standards	5
3.1	Coverage Standards	5
3.1.1	Functional Test Cases	5
3.1.2	Structural Test Cases	5
3.2	Analyses	5
3.2.1	Test Suite Construction	5
3.2.2	Incremental Test Analyses	5
4	Modifying the Unit Test Plan	7
4.1	Test Effectiveness	7
5	References and Further Reading	9

Deleted:	1 . Overview	1¶
	1.1 Identification	1¶
	1.2 Document Map	1¶
	1.3 Concepts	2¶
	1.4 Readership	2¶
	2 Template for the Test Plan	3¶
	2.1 Test Suspension Criteria and Resumption Requirements	3¶
	2.2 Problem Determination and Correction Responsibilities	4¶
	3 Analyses and Standards	5¶
	3.1 Coverage Standards	5¶
	3.1.1 Functional Test Cases	5¶
	3.1.2 Structural Test Cases	5¶
	3.2 Analyses	5¶
	3.2.1 Test Suite Construction	5¶
	3.2.2 Incremental Test Analyses	5¶
	4 Modifying the Unit Test Plan	7¶
	4.1 Test Effectiveness	7¶
	5 References and Further Reading	9¶

Formatted: Do not check spelling or grammar

List of Figures

Figure 1: Document Map	1
Figure 2: Template for the Test Plan	3
Figure 3: Attached Process	7

[.....](#)

Deleted: [Figure 1: Document Map . 1](#)
[Figure 2: Template for the Test Plan . 3](#)
[Figure 3: Attached Process . 7](#)

1 Overview

1.1 Identification

The Arcade Game Maker (AGM) product line organization will produce a series of arcade games ranging from low obstacle count to high with a range of interaction effects. More detailed information can be found in *Arcade Game Maker Pedagogical Product Line: Scope*. This document describes how individual code assets are tested in the AGM product line.

1.2 Document Map

The AGM product line is described in a series of documents. These documents are related to each other as shown in Figure 1. This map shows the order in which the documents should be read for the first time. After readers are familiar with the documents, they can go directly to the information needed.

This document is the unit test plan template. Product line organizations use this document to capture how each code unit is tested. This document follows the outline provided in the report by McGregor [McGregor 01b].

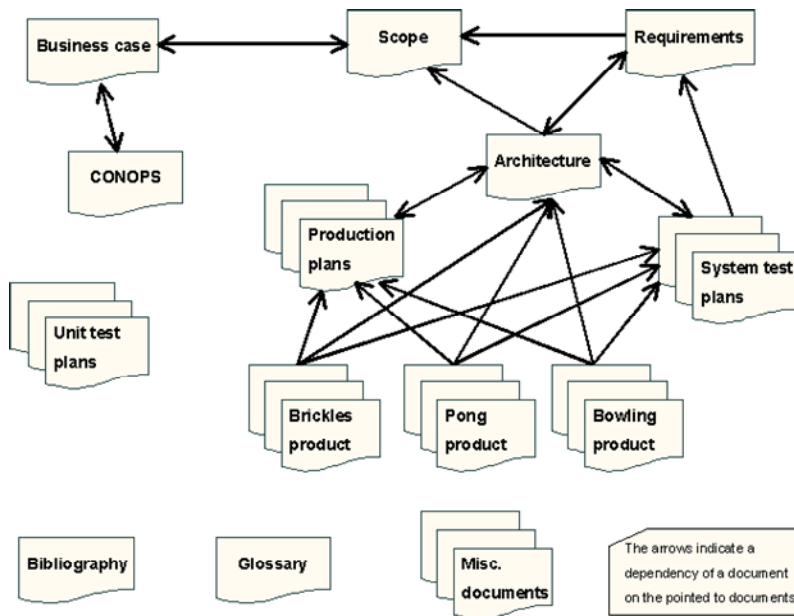


Figure 1: Document Map

1.3 Concepts

For definitions of basic concepts, see *Arcade Game Maker Pedagogical Product Line: Glossary*.

1.4 Readership

This document is intended to provide some information to all of the stakeholders in the Arcade Game Maker framework, but it is intended primarily for core asset development teams. The unit test plan describes for a manager the resources that are needed to test an asset. Technical members of the organization can use the unit test plan template as a guide for developing a complete unit-specific test plan.

2 Template for the Test Plan

The AGM product line organization has decided to follow the Institute of Electrical and Electronics Engineers (IEEE) 829 standard for test plans. The template shown in Figure 2 is filled in for each unit. See the report by McGregor [McGregor 01b] for definitions of each of the following sections:

1. Introduction
2. Test Items
3. Tested Features
4. Features Not Tested (per cycle)
5. Testing Strategy and Approach
 - 5.1 Syntax
 - 5.2 Description of Functionality
 - 5.3 Arguments for Tests
 - 5.4 Expected Output
 - 5.5 Specific Exclusions
 - 5.6 Dependencies
 - 5.7 Success/Failure Criteria for Test Cases
6. Pass/Fail Criteria for the Complete Test Cycle
7. Entrance and Exit Criteria

Increment	Functionality	Schedule
1		
2		
3		

Figure 2: Template for the Test Plan

The unit test phase runs concurrently with the unit development phase. The construction of test cases begins while the unit's specification is being developed. Feedback from test-case development allows the developer to identify and correct inconsistencies and ambiguities in the specification.

The unit test phase is exited when the developer has implemented the required unit and it has passed all of the tests required by the standards defined in the test plan template.

2.1 Test Suspension Criteria and Resumption Requirements

Unit tests are suspended for one of two reasons:

1. The available functionality has been adequately tested and has passed those tests, but additional functionality remains to be developed.
2. The available functionality has not passed the tests, and the developer has sufficient information to make another development pass.

Tests are resumed when the developer has constructed additional functionality or revised the existing functionality.

1. Test Deliverables/Status Communications Vehicles
2. Testing Tasks
3. Hardware and Software Requirements

No special hardware is required for the tests of this cluster.

The DotUnit executable and the accompanying framework classes are required for building test classes.

2.2 Problem Determination and Correction Responsibilities

The developer has responsibility for corrections. The users of the cluster are responsible for reporting any problems to the listed owner of the cluster.

1. Staffing and Training Needs/Assignments
2. Test Schedules
3. Risks and Contingencies

Increment	Functionality	Schedule

3 Analyses and Standards

This section describes the techniques and the agreed-upon standards used to test each component in the AGM product line. These are justified and put into context in the report by McGregor [McGregor 01b].

3.1 Coverage Standards

3.1.1 Functional Test Cases

For each service on a component, a test case is constructed for every clause of the post-condition.

The unit invariant is tested before and after each service invocation. This testing can be done in conjunction with the service test cases.

3.1.2 Structural Test Cases

A test case is constructed for each sequence of statements. This effort can be optimized by first executing the functional tests while running a code-coverage tool. Then, it is necessary to construct only test cases to cover sequences of statements that were not covered by the functional test cases. Be certain that this test case includes all exceptional sequences.

3.2 Analyses

3.2.1 Test Suite Construction

The AGM product line organization has decided to use the test-case selection techniques described in the report by McGregor [McGregor 01b]. The information in that report should be read before constructing test suites.

3.2.2 Incremental Test Analyses

After the initial test suites have been created, different techniques are used to maintain the test suites. These techniques should be applied every time the unit is changed.

Change Impact Analysis. Use the results from the change impact analysis that was conducted by the developers. That analysis will have identified those portions of the asset that will be modified when the change is implemented.

Diff. A tool such as diff can be used to show the exact difference between two versions of the unit. Then, tests can be modified to address just those differences.

4 Modifying the Unit Test Plan

A specific unit test plan is modified every time the unit being tested is changed using the techniques discussed in Section 3.2. The generic unit test plan is modified when it is shown that the techniques are not producing effective test cases and the standards are not producing satisfactory results. That modification process is the focus of the attached process shown in Figure 3.

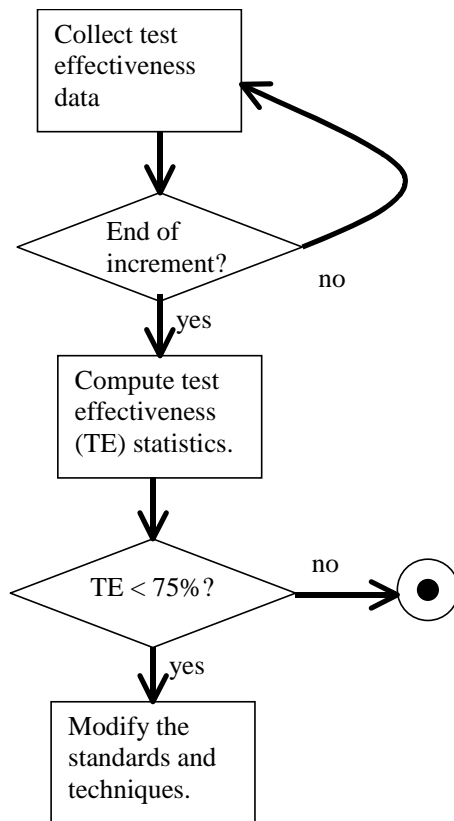


Figure 3: Attached Process

4.1 Test Effectiveness

Test effectiveness (TE) is measured by the number of defects that escape detection by the tests that are run. For the unit test phase, any defect found in a component after unit test counts against the TE of the unit test. Test effectiveness is computed as follows:

$$TE = \frac{TotalDefects - DefectsFoundAfterTest}{TotalDefects}$$

Defects are cataloged as they are identified and analyzed to determine their origin. At the end of a product line increment, the test effectiveness is computed on a component-by-component basis. When the average TE goes below 75%, the test coverage standards are made more comprehensive.

5 References and Further Reading

For references, see *Arcade Game Maker Pedagogical Product Line: Bibliography*.