

Systems Analysis and Design

10 Testing

Terry Marris December 2007

The purpose of testing is to find errors. We see how to construct test cases and test plans.

10.1 Where Errors are Found

Errors may occur at any stage of the system life cycle:

- in the analysis
- in the specification
- in the design and
- in the implementation.

Errors that are not noticed in the early stages of the life cycle get passed on to the later stages and become more difficult and costly to put right. In this respect, the analysis and the specification are far more important than the design and the implementation.

We start with the use cases because they tell the customer what to expect, the programmer what to code, the user-guide author what to write and the tester what to test.

10.2 Test Case

A test case describes how to carry out a particular test. We start with a use case.

Use Case: Loan

Purpose: to record the loan of a copy to a borrower

Actor: librarian

Main Success Scenario

- 1 librarian presents borrowers id
- 2 system finds borrowers record
- 3 librarian presents copy id
- 4 system finds copy record
- 5 system adds copy id and date due for return to borrowers record
- 6 exit success

Exceptions

- 2a borrower not found
 - 2a1 exit failure

- 2b borrower has reached lending limit
 - 2b1 exit failure

- 2c borrower has fines not yet paid
 - 2c1 exit failure

- 4a copy not from loan stock
 - 4a1 exit failure

A use case has several scenarios: main success scenario and various exception (or error) scenarios. Each scenario generates a test case. A Test Plan is a collection of test cases.

Test Plan for Use Case: Loan

Test Case	Scenario	borrowerId	copyId	lendingLimit Reached	outstanding FineOwed	expected Result
L1	Success	Valid	Valid	No	No	loan of copy to borrower recorded
L2	Borrower not found	Invalid				Error: borrower not found
L3	Lending limit reached	Valid		Yes		Error: lending limit reached
L4	Unpaid fines	Valid		No	Yes	Error: unpaid fines
L5	Copy not loanable	Valid	Invalid			Error: copy not for loan

Notes:

1. The Loan use case has generated five test cases.
2. Each test case is uniquely identified e.g. L1 (L for Loan).
3. A scenario is either successful, or an error for some reason.
4. borrowerId is either valid - it is in the file of borrower records, or invalid - it is not in the file of borrower records.
5. copyId is either valid - it represents a loanable copy, or invalid - it does not represent a loanable copy.
6. lendingLimitReached is either true (yes) or false (no).
7. outstandingFineOwed is either true (yes) or false (no).

Test plans are developed, added to and refined, as you progress through the analysis and design. For example, after completing a data dictionary you can replace the Valid, Invalid, No and Yes entries with actual values.

Systems analysis is not a write once and forget it exercise; it is an exercise of continuous development and improvement.

Bibliography

FOWLER M *UML Distilled* 2nd Ed Addison-Wesley 2000

<http://catless.ncl.ac.uk/risks> accessed 20 Dec 2007

http://www.therationaledge.com/content/jun_01/m_cases_jh.html accessed 20 Dec 2007