

Developing a Risk-Based Test Strategy

With the rapid pace of application development in the e-business world, testing has become a challenging proposition. Trying to meet even tighter deadlines while still delivering products that meet customer requirements is the greatest challenge testers face today. Formulating answers to age-old questions like “What should we test?” and “How long do we test?” requires different strategies in fast-paced environments.

In today’s application development environments, stakeholders are looking for testers to provide answers to additional questions such as:

- Does the product meet our quality expectations?
- Is the application ready for users?
- What can we expect when 2,000 people hit the site?
- What are we risking if we release now?

Meeting the Demands

One way for testers to meet the demands of rapid application implementation is to use a risk-based approach to defining requirements and strategy. Such an approach allows you to assess the risks of potential problems in the product compared to the quality expectations that a stakeholder has. A sound risk-based test strategy can increase the probability that:

- The most important problems are found
- Problems are detected early
- Problems with the most potential for re-work are found first
- Requirements with the most impact to users are tested first
- Accurate information on product quality can be provided

The proposed risk strategy for testing moves us from the informal approach experienced testers often use to a more formal and systematic way of assessing risk. This allows you to base your test strategy on the assessment, as well as address the quality concerns of the stakeholder.

Assessing Product Risk

According to software risk management, risk is defined as “the possibility of failure times the potential damage if the risk occurs.” The risk assessment proposed is a simplified version of more formal risk assessment techniques that project managers often use. Fourteen of the most common risks identified by best practice literature are listed. The goal is to assess the risk for each high-level module/feature within a product. The tester reviews each listed risk and assesses the likelihood of that risk impacting the particular module/feature under consideration.

The following step-by-step instructions provide a guide to assessing product risk and calculating the risk score. A simple scale of low to high is used to rate the probability of that risk occurring for each feature. After each risk is rated, a risk score for each module or functionality is calculated.

1. Use the following scale when assessing risk for each module:

- N/A (0) - This risk doesn't apply to this module or function
- Low (1) - This risk could occur, but it isn't likely
- Medium (3) - This risk probably will affect this module or function
- High (5) - This risk definitely will impact this module or function

2. Rate each functional unit or module in the proposed system based on the likelihood of that risk occurring.

3. Calculate a risk score for each module.

4. Rank the modules, starting with the highest risk score.

The table below provides an example of a risk tool and a completed assessment of four software modules (functions 1-4). The tool can be customized to include different risks for specific application types such as e-commerce or object-oriented software projects.

Product Risk Assessment

Potential Risk Function	1 Function	2 Function	3 Function	4 Product
1. Modules that will be used heavily by users	High (5)	N/A (0)	Medium (3)	High (5)
2. Modules with very complex functions	High (5)	High (5)	Low (1)	Medium (3)
3. Modules that have been "fixed" or updated often	Low (1)	Low (1)	Low (1)	Low (1)
4. Functions that require high availability	N/A (0)	N/A (0)	N/A (0)	N/A (0)
5. Functions that require consistent performance levels	Medium (3)	N/A (0)	Low (1)	Medium (3)
6. Functions that use new development tools & languages	N/A (0)	N/A (0)	N/A (0)	N/A (0)
7. Functions with many interfaces	N/A (0)	N/A (0)	Low (1)	Medium (3)
8. Functions developed by inexperienced developers	N/A (0)	N/A (0)	N/A (0)	N/A (0)
9. Functions developed with inadequate user involvement	High (5)	High (5)	High (5)	High (5)
10. Functions developed by a large development team	High (5)	High (5)	High (5)	High (5)
11. Completely new functions	High (5)	N/A (0)	N/A (0)	High (5)
12. Functions developed under extreme time pressure	Medium (3)	Low (1)	Low (1)	High (5)
13. Most important function to stakeholders	Medium (3)	Low (1)	Medium (3)	High (5)
14. Functions that had large numbers of defects in previous versions	N/A (0)	N/A (0)	Low (1)	High (5)
Risk Score	35	18	22	45

Developing a Test Strategy

A risk-based test strategy provides the means for identifying the optimal balance between the testing effort and the amount of product testing needed to lessen the impact of the identified risks. Your risk-based strategy should describe what will be tested, how it will be tested and how long it will be tested. A review of the risk assessment results allows the tester to determine:

- The order for testing system components
- The modules that may require more test coverage than others
- The types of testing performed for each module and the system

A test strategy based on our risk assessment would begin with ranking the components using the risk score calculated during the risk assessment. In the example above, the module with the highest risk would be module 4, followed by modules 1, 3 and 2. Modules with the highest risk ranking require more testing than modules with lower scores. The overall risk ranking helps the tester determine which modules require the most test coverage. These modules should consequently be tested first and for longer periods of time.

A review of the specific risks for each module assists the tester in determining the specific tests that should be performed for testing. For example, a module that has performance and availability risks should be scheduled for additional tests beyond straight functional testing. Modules that are at risk for usability would require additional tests in that area.

Many test managers have intuitively used information about testing risks to help make testing decisions. This risk assessment process provides a more formal way for test managers to utilize business risks to determine testing priorities. Developing a risk-based test strategy early in the project can help organizations address all the critical questions for successful application implementation, particularly when time is compressed.

Visit us at www.spherion.com to learn how your entire organization can optimize your migration to an updated operating system, enhancing productivity and end-user satisfaction throughout your organization.