Gain Precision Effectiveness with Risk-Based Testing

Executive Summary

In today's fiercely competitive world, customers want high-quality software, but they want it quickly and at low cost. Faced with these pressures, some organizations try to rush testing, which endangers quality and increases long-term support costs. Other organizations over-test in hopes of catching every defect, which wastes money and can delay software's delivery to the market.

Organizations must strike a balance between reducing overall project costs and test schedules, while assuring quality.

To do so, they need a consistent, business-driven process to decide which functions and requirements are most important to test, and in what order. Risk-based testing (RBT) delivers this capability by assessing the likelihood of defects, as well as the business impact if any specific defect causes a failure in the software. But RBT is only effective if organizations assess those risks properly and consistently, with the proper input from their business partners.

Cognizant's Risk Inspired Scenario Evaluator (RISE), an internally developed RBT methodology, reduces the cost and time required for testing, while improving quality and assuring that testing is focused on the most critical areas. RISE does a more precise job of assessing risks than other methodologies because it requires subject matter experts and business users to assign a quantitative value, rather than a less precise qualitative measure, to the likelihood of a defect and to the business impact of a resulting failure. It is also flexible, allowing any number of risk factors to be included in the risk calculation. This formal, well-defined and quantitative ranking is better than ad hoc, informal and subjective approaches used to balance quality with schedule and budget limitations.

Testing Challenges

In a perfect world, software testing organizations could perform exhaustive quality assurance (QA) to assure every feature and function performed perfectly under all conditions. In the real world, they must trim testing schedules to outmaneuver competitors or to help users quickly increase their productivity, reduce costs or improve customer service. They must also perform testing within strict budget limits.

Test organizations are also often hampered by a lack of skilled testers, due in part to tight budgets and rushed schedules. As a result, the test team might be unfamiliar with the application being tested, the business domain in which it will operate, the technical environment (such as mobile platforms) on which it will run or even proper test processes.

Since it's impossible to effectively test every part of every application, organizations struggle to decide how much testing to do on which application modules, and in the proper sequence, to meet customer needs at the lowest possible cost. Failing to prioritize these test efforts properly results in the following outcomes:

- Not finding critical defects until late in the development cycle, which significantly increases the time, and cost, required to fix them.
- Wasting time and money testing less important parts of the application that contain fewer likely defects.
- Higher software testing and remediation costs, by testing a larger number of test cases than are necessary.
- Delays in delivering software to internal or external customers.

Unsure of how much testing any particular application needs, and of where the most dangerous defects might be, organizations are faced with a number of unsavory alternatives. One is to simply reduce the scope of testing. But without a way to prioritize test needs, this risks missing critical defects.

A second (if the organization can afford it) is to devote more testers and budget to the testing. But without proper prioritization, this doesn't assure the testing will find the most critical defects. In any case, spending more money than necessary puts the organization at a competitive disadvantage.

The third is test case optimization, which attempts to predict, based on the complexity of a unit of code, the likelihood and criticality of the defects in it. Because this prioritization is based only on technical rather than business considerations, it can waste test efforts on modules that have little effect on the business. Another limiting factor is that such prioritization is performed only on test cases, which are themselves subsets of a business scenario (such as entering an order or underwriting an insurance policy). As a result, it fails to estimate the impact of a failure on the business. Finally, test case optimization is so complex and difficult that it can increase test costs while delivering little benefit.

Test automation, the fourth option, would seem to be a cure-all, but is not effective at reducing costs for all test cases. Because it involves high up-front costs, an organization must also be sure test automation tools and processes can be used over time to assure a return on their investment.

A far more effective approach, described in this paper, is RBT, which examines at the level of the business scenarios the likelihood of a given defect and the business impact of such a defect. This helps assure companies spend the most time and money on the most critical areas, and find the largest number of defects (and the most important defects) in the shortest time and at the lowest possible cost.

Bringing the right level of consistency and business input to prioritizing test efforts is what sets Cognizant's RISE apart from other RBT methodologies.

This white paper describes the advantages of this methodology and the critical factors in implementing it correctly.

RBT Done Right

RBT is the process of prioritizing what software to test based on:

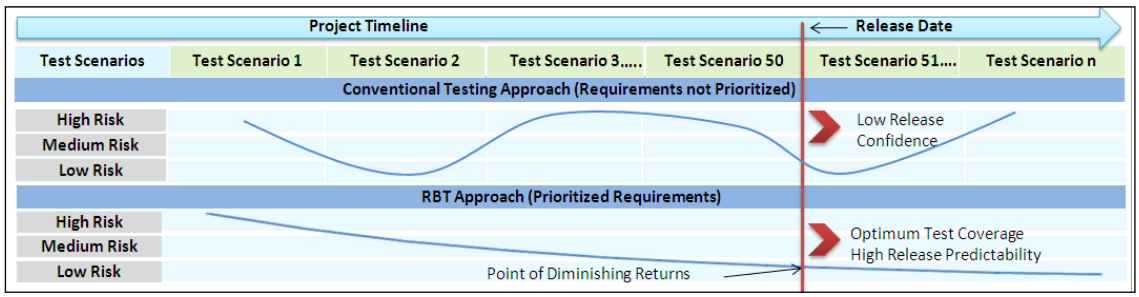
- The mathematical risk of the failure of a given software module if it is not tested.
- The impact on the business of such a failure.

Based on this analysis, project managers can focus testing on the most important scenarios or cases, selecting the amount of testing to do based on their project constraints and the amount of risk the company can afford to take.

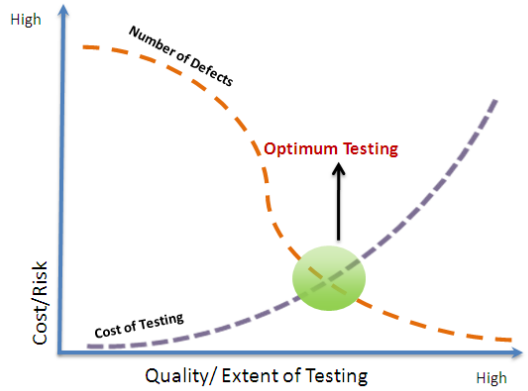
A properly implemented RBT methodology helps assure that high-risk areas are tested first, then medium-risk and, finally, low-risk areas.

This is an improvement over non-prioritized testing, in which low-risk areas might be tested before higher-risk modules (see Figures 1 and 2, below).

Dueling Approaches: The RBT Advantage



When RBT is done right, test dollars and staff are focused on the most business-critical areas, with test efforts stopping when they reach the point of diminishing returns.



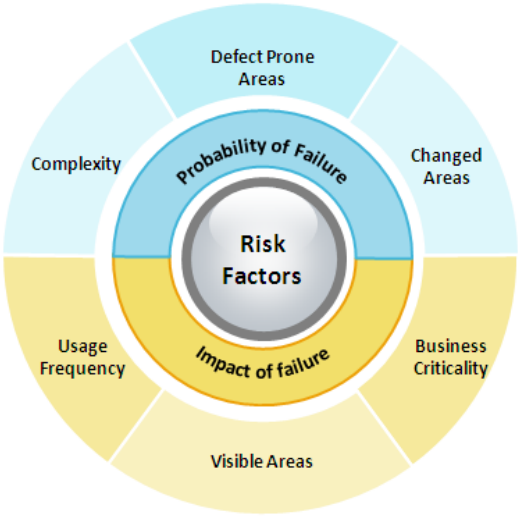
Obviously, this process works well only when the risks are properly prioritized. Cognizant's RISE methodology is designed to capture risks at a very detailed level, and to capture the appropriate quantitative input from business and subject matter experts. It is these experts who are best able to estimate both the likelihood of a defect and the business impact of such a defect.

The process begins with the preparation of a strategy describing the various phases of the RBT process, including the risks and critical success factors. With a strategy in place, the quality assurance (QA) manager incorporates feedback from the business and IT managers, as well as the SMEs to set the prioritization guidelines. These guidelines are essential for consistent and accurate risk evaluation and will be the reference point for test analysts as they assign risk ratings to each requirement.

The next process analyzes the risk of both the likelihood of defects and the effect of any defect on the business. This risk analysis is performed at two levels. The first is at the level of individual functionalities, or modules, within the application (such as order entry or billing). The second is performed at the level of each requirement within a module (such as "validate customer credit limit" within the "order entry" module). The combination of the functional and requirement risk analyses produces a map of the required test coverage, as shown in Figure 3, below.

Because risk analysis is so important, RISE captures feedback about risks in a well-defined, granular and quantitative process. The first step is the RBT workshop, in which the test analysts, business users and subject matter experts (business analysts) identify the various risk factors and assign weights to each. For example, if an application undergoes frequent changes, the risk factor "Changed Areas" will be given a greater weight. On the other hand, if an application is not used directly by end users but only by another application or process, the risk factor "Visible Areas" (the functional areas of an application seen by users) may be given little weight, or not even included as a parameter.

Assessing Risk



The RISE process prioritizes testing by "must have," "important" and "nice to have." For example, a "must have" requirement in an "important" functionality will be included in test suite.

Risk Exposure= (Probability of Failure [1] * Impact of Failure [2])

[1]Probability of Failure defines the likelihood of a particular functionality failing.

$$\text{Probability of Failure} = \frac{\sum(\text{Probability Risk Factor}_{\text{Weightage}} \times \text{Probability Risk Factor}_{\text{Value}})}{\sum(\text{Probability Risk Factor}_{\text{Weightage}})}$$

[2]Impact of Failure defines the impact of not detecting the failure.

$$\text{Impact of Failure} = \frac{\sum(\text{Impact Risk Factor}_{\text{Weightage}} \times \text{Impact Risk Factor}_{\text{Value}})}{\sum(\text{Impact Risk Factor}_{\text{Weightage}})}$$

1. **Probability of Failure=**
*[{(Defect Prone Areas Weightage * Defect Prone Areas Value Functionality n) + (Changed Areas Weightage* Changed Areas Value Functionality n) + (Complexity Weightage* Complexity Value Functionality n)}/ (Defect Prone Areas Weightage + Changed Areas Weightage + Complexity Weightage)]*

2. **Impact of Failure=**
[{(Usage Frequency Weightage Usage Frequency Value Functionality n) + (Visible Areas Weightage* Visible Areas Value Functionality n) + (Business Criticality Weightage * Business Criticality Value Functionality n)}/ (Usage Frequency Weightage + Visible Areas Weightage + Business Criticality Weightage)]*

The next step is a "crowd rating" in which users and business analysts analyze the various functional modules and provide a quantitative rating of the risk that a specific module will contain defects, and the business impact caused by the failure of that module. This assures the test team will assign a higher level of importance, and perform more comprehensive testing, on the specific functional areas of the application the business considers most important.

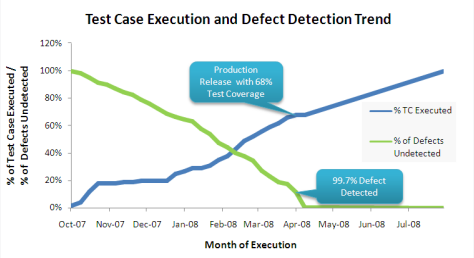
The second risk analysis is done at the requirement level. This begins with adding the risk parameters and their weights (developed in the RBT workshop) to Cognizant's RISE tool. The next step is to calculate the effort required to test every application, and the number of test cases that can be created or executed in a day, to estimate the total required test effort. This can be used to compare the estimated test effort required using RBT to that of exhaustive, end-to-end testing.

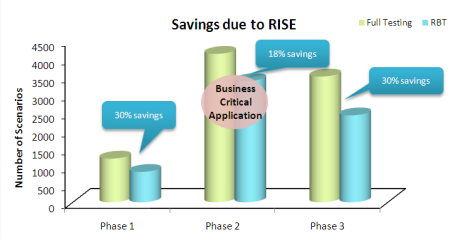
The requirements are then imported into RISE, and the risk factor for each is graded on a numeric scale. Next, the requirements are sorted according to the risk exposure and classified as "must have," "important" and "nice to have" (see Figure 4).

Making the Grade

RBT At Work

We used our unique risk-based testing process to help a large U.S. insurer meet strict time constraints for modernizing a major application suite, while reducing the cost of testing despite a limited pool of domain testers. The use of RISE helped the insurer find 99.7% of defects, while cutting the cost of testing by 30%, depending on the number of business-critical modules in a testing phase. Our RISE methodology helped the insurer meet a strict deadline that allowed only seven to eight months for testing. Among other benefits, the use of our risk-based testing methodology reduced the costs associated with late defect detection.





Our RISE methodology enabled the insurer to prioritize testing and ensure that a higher percentage of scenarios are tested for the most critical parts of the application.

After the risk analysis, the exact test coverage can be determined manually based on project constraints such as time and cost, or through the use of a test coverage optimization model.

The RISE process prioritizes testing by "must have," "important" and "nice to have." For example, a "must have" requirement in an "important" functionality will be included in test suite.

Determining the test coverage manually can be a good option if the team has experience with similar projects. This can help the team determine the appropriate level of test coverage (assuming they verify that the level of coverage meets the project's time and cost constraints). An automated model determines the optimum coverage based on parameters such as defect density, error discovery rate, quality of fixes and the number of post-production defects based on historic data from previous projects.

The model calculates the AQI (Application Quality Index) corresponding to various levels of test coverage ranging from 0% to 100%, and recommends an optimum level of test coverage to produce the lowest AQI that certifies the application as ready for deployment. (Cognizant's proprietary Application Quality Index uses 12 parameters to determine application quality at any point during testing, and helps managers decide whether it is ready for production.) Once the appropriate test coverage is determined, testing is performed, and an analysis is done of defects found during system testing, user acceptance testing and production use. This analysis is used to reassess the effectiveness of the risk parameters and adjust them, if necessary, to better optimize future testing.

Once risk-based testing is complete, an implementation assessment should be done to measure critical metrics such as defect density, error discovery rate, quality of fixes and the number of post-production defects. These metrics are used to fine-tune the model so it can more accurately determine proper test levels in the future.

RBT At Work

Proper risk assessment is the very core of an RBT effort and requires that the following factors exist:

- Quality analysts (the testers performing the risk evaluation) have the proper level of expertise in both the business domain and the application.
- Business users, subject matter experts (business analysts) and QA teams collaborate effectively in choosing the modules to be tested in the final test suite.
- The test team makes the proper use of historical data, such as which modules have been historically error prone.
- The necessary business users and subject matter experts be available for the RBT workshops.
- Business analysts and the test team communicate effectively so they have a good, mutual understanding of the scope of the test suite. Often, business analysts and testers have different priorities and time constraints, and clear communication is required to ensure both sides can properly coordinate the test process.

Conclusion

Organizations have always had to prioritize their software testing efforts, but the need to properly decide what to test, and when to test it, is greater than ever. To compete, enterprises must cut testing costs as far as possible, without sacrificing quality of application or delaying critical application launch time. Risk-based testing is the most effective way to assure that only the most critical units of software are tested, and that they are tested in the proper order to minimize delay and costs, and to assure the greatest business benefit.

Based on our work with enterprises worldwide, we have developed a unique, quantifiable risk-based testing process backed by statistical techniques and proven testing best practices. This process assures the appropriate amount of input from business and technical experts, and the proper level of detail in prioritizing risks. This gives businesses a consistent, business-based methodology for determining what level of software testing will deliver business value and the highest possible customer satisfaction.

About Cognizant : Cognizant (Nasdaq: CTSH) is a leading provider of information technology, consulting, and business process outsourcing services. Cognizant's single-minded passion is to dedicate our global technology and innovation know-how, our industry expertise and worldwide resources to working together with clients to make their businesses stronger. With over 50 global delivery centers and more than 85,500 employees as of March 31, 2010, we combine a unique onsite/offshore delivery model infused with a distinct culture of customer satisfaction. A member of the NASDAQ-100 Index and S&P 500 Index, Cognizant is a Forbes Global 2000 company and a member of the Fortune 1000 and is ranked among the top information technology companies in BusinessWeek's Hot Growth and Top 50 Performers listings. Visit us online at www.cognizant.com

Cognizant | Testing Services
Passion for building stronger businesses

World Headquarters
500 Frank W. Burr Blvd.
Teaneck, NJ 07666 USA
Phone: +1 201 801 0233
Fax: +1 201 801 0243
Toll Free: +1 888 937 3277
Email: inquiry@cognizant.com

European Headquarters
Haymarket House
28-29 Haymarket
London SW1Y 4SP UK
Phone: +44 (0) 20 7321 4888
Fax: +44 (0) 20 7321 4890
Email: infouk@cognizant.com

India Operations Headquarters
#5/535, Old Mahabalipuram Road
Okkiyam Pettai, Thoraipakkam
Chennai, 600 096 India
Phone: +91 (0) 44 4209 6000
Fax: +91 (0) 44 4209 6060
Email: inquiryindia@cognizant.com