Risk Based Testing in the Digital Age

These days there are ample projects, more competitive pressures and greater failure risk which must be managed often with fewer resources and tighter schedules. Especially in the current age of digital revolution, mobile applications must be tested on various mobile devices and operating systems under different network conditions. There are endless permutations and combinations of tests to perform but at the same time there is pressure of reducing time to market.

But with all these limitations, there's simply no room for compromise on quality and stability especially in the case of important business critical applications. So, rather than doing more with less and risking delayed projects, increased costs or low quality, we must find ways to achieve better with less. The focus of testing must be on aspects of software that matter most with a goal of bringing down the risk of failure as well as ensuring the quality and stability of the applications. This can be accomplished by applying the principle of Risk Based Prioritization of tests, known as Risk-based testing (RBT).

In a general testing approach, test prioritization is left mostly to Quality Engineer’s discretion and is based on the tacit knowledge. High-defect pipeline does not necessarily indicate that all defects being identified internally are of high value. Also, fixing the issues in the order of internal severity may be flawed from a customer perspective. Typically, testers tend to give equal weightage to all enhancements during release test cycles as opposed to risk-based weightage. RBT enables testers to assess the Risk from various dimensions with insights from stakeholders including Quality Engineers, Coders, Business Analysts and Product Managers.
What is Risk-Based Testing?

Broadly, the Risk Based Testing strategy consists of 3 steps:

a) Identify and Assess the risk across all modules for the product
b) Develop a Test Strategy
c) Implement the Test strategy

- Focus on Tests based on Risks
- Thorough understanding of:
  - Cost of failure
  - Probability of failures
  - Defect Density
- Informed decision making
- Right prioritization of Test Execution

Risk Based Testing strategy helps to focus on Test Execution based on Risk level across modules. It enables assessing risk from a Cost and Probability of Failure perspective while also considering the
Defect Density across modules. It guides in making informed decisions on the right prioritization of Test Execution.

With **Risk** being the key word here, let’s understand what is Risk and how do we determine its weightage.

**The Science of Risk Calculation**

Overall Risk is calculated as a product of Cost of Failure and Probability of Failure.

---

**The Science of Risk Calculation**

“Risk-based testing (RBT) is a testing strategy that functions as an organizational principle used to prioritize the tests of features and functions in software, based on the risk of failure, the function of their importance and likelihood or impact of failure.”

---

**Cost of Failure** is the extent of damage or financial impact a system failure can result in. It is typically viewed from the business perspective and hence assessed based on few further parameters such as Business Criticality, Usage Frequency, and Visibility.
**Probability of Failure** is the likelihood of failure in a system, feature, or a process. Probability of Failure is assessed based on new technology or new functionality getting introduced into the system, or any Design Optimization being included. Couple of other parameters that drive this are the Change Frequency and Complexity in terms of functional/technical design.

We will be discussing these terms in detail further.

Risk Assessment exercise yields a chart (plotted with the help of an excel macro) that distributes the modules across High, Medium, and Low Risk levels represented as Red, Yellow, and Green zones respectively. Any failures in the modules falling in Red zone have high cost of failure and high probably of failure. They must be tested with the highest priority with maximum test coverage.

In a situation where one cannot address all modules in medium risk level due to capacity constraints, Test coverage depth for modules in this area must be planned with an intent to reduce the extent of failures.

Similarly, Test coverage for modules in Low risk level must be planned with a goal to limit the failures in these modules.

At the least, modules falling in Medium and Low risk areas must get sanity tests done to ensure no basic functionality is broken.

Before getting into the definition of individual parameters, one should be acquainted with the terminologies of Weightage and Scale.

**Weightage**: Based on the industry standards relative weightages are assigned across the Risk Factors to differentiate the relative importance. Business Criticality, New Technology, and New Functionality/Design Optimization are given the highest Weightage of 20 while the Usage Frequency and Change Frequency are rated at 10.
Visibility and Complexity are given the lowest Weightage of 5. These Weightages are fixed and should not be altered during the process of Risk Assessment. The idea behind having distinctive weightages is to enable clear demarcation of final computed risk values.

**Scale:** A Scale of 1, 4, and 7 is used to reflect the corresponding Low, Medium, and High degree of influence across the risk factors. For every chosen module for Risk Assessment, a suitable Scale (1 or 4 or 7) must be chosen across risk factors. Risk Assessment is done as a combined effort from Quality Engineer, Coder, Business Analyst, Product Manager and Design Lead.

Having defined both Weightage and Scale, let’s deep dive into the definition of these parameters.

**Business Criticality** refers to any factor of a system whose failure will result in the failure of Business Operations. Based on the nature of failure a firm could suffer financial, legal, or other damages.

The impact of Business Criticality may vary based on the nature of failure.

It’s a best practice to compute Business Criticality cost aspect with inputs from Business Analyst, Product Manager and Design Leads.

**Usage Frequency**

Next, we would be looking at the 2nd parameter influencing the cost of failure, Usage Frequency.

Usage frequency is the number of times a functionality is being used in a given span of time by a customer or the number of customers using a functionality.

This information can be collected with the help of the Business Analyst and/or design leads who have exposure to client operations.
**Visibility**

Now, let us talk about the last parameter influencing the cost of failure, Visibility.

Visibility is the degree to which something is seen or known about.

In other words, Visibility would tell us the degree to which an issue, if arising in a module, impacts or is noticed by the users across the breadth and depth of the organization irrespective of the criticality.

With that we have covered all the parameters for Cost of Failure. Let’s move on to the parameters influencing the Probability of Failure.

**New Technology**

First in the list is New Technology.

Probability of failure due to new technology can be defined as Software failures when one technology or better, a set of technologies are replaced by another technology and/or set of technologies.

In other words it would refer to the probability of a failure caused by any code or technology change in a particular functionality. These changes may not have a direct relation to the business logic of the application, instead they are purely technical in nature.

**New Functionality/Design Optimization**

Now let us move on to the 2nd parameter influencing the Probability of Failure, New Functionality or Design Optimization.

The probability of failure due to new functionality or design optimization can be defined as the probability of a software failure either due to a functional enhancement or a design optimization.
**Change Frequency**

The Change frequency can be defined as the impact of the rate at which a specific code area is changed to resolve Customer Identified Issues (CII) or Internal Defects.

The weightage for change frequency must be derived based on the frequency of code change and the CII or defect trend over a fixed time frame.

In other words, the change frequency should address the impact of the magnitude of the code change by a CII or defect along with the nature and/or priority of the defects.

**Complexity**

The last parameter influencing the Probability of Failure is Complexity.

The complexity of a functional area refers to the design, technical as well as functional complexity. The complexity factor when considered along with the code changes would provide more clarity on the probability of failure.

**Weightage and Scale**

Below picture shows the summary of Risk Parameters, their Weightage, and related Scales which were discussed in the preceding sections. Total Cost is calculated by multiplying the Weightage with the corresponding Scale value and summing up the values across the Cost parameters.
Likewise, the Probability of Failure is also calculated and finally the Cost of Failure and Probability of Failure are multiplied to arrive at the overall Risk Weightage.

**Risk Ranking**

Let us move on to the Risk Ranking and the reasoning for the same. As you can see, the chart is divided into three sections. The areas marked in red pose the highest risk and hence are a must to be addressed areas. The areas marked yellow and green are next in line in the descending order of priority from the risk perspective.
Within these high-level categorization of risk areas, we need a prioritization of the areas for planning the test strategy.

While ranking within the Risk chart, more priority is given to the cost of failure. As you can see in the chart, the highest priority would go to the Most certain and Catastrophic risk areas. Any module or transaction that falls under this needs to be considered as part of the testing scope with the highest priority.

Following the same pattern, the next in priority would be the Likely and Catastrophic area as it could incur a very high cost on any issue that crops up in modules that fall in this area. The third priority as suggested needs to go to the Almost certain and Major area. The areas of major cost can incur a priority 1 issue and if the probability of the issue is almost certain, this area becomes an area of risk. We would suggest this area to be considered as the next in priority area for testing scope.

The areas of risk marked 4, 5, and 6 follow the original pattern based on the cost of failure.
In the area marked as Yellow which falls under the next category of priority, the priorities are first ranked based on the cost as you can see for 7, 8, 9, and 10 risk areas. The rest of them are based on the probability as now we have lesser cost involved in these areas.

Another aspect that we may need to address is the prioritization of the modules or transactions that fall under the same risk area. For Example, two transactions that fall under the risk area 7, where we are in need to choose one between them to add to the scope of testing. In such a situation, the team is advised to consider the absolute risk value which is the product of the cost and probability of failure.

With this exercise, we will arrive at a test scope for the release or a testing activity planned midway through the release, which is more rational in approach and would yield high value defects and eliminate risks in the product.

**Sample illustration of Risk Assessment and the resulting reports**

**Risk assessment for 10 transactions**

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Transaction 1</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>385</td>
<td>245</td>
<td>25</td>
</tr>
<tr>
<td>Transaction 2</td>
<td>7</td>
<td>4</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>265</td>
<td>235</td>
<td>24</td>
</tr>
<tr>
<td>Transaction 3</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>7</td>
<td>175</td>
<td>245</td>
<td>22</td>
</tr>
<tr>
<td>Transaction 4</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>7</td>
<td>175</td>
<td>245</td>
<td>22</td>
</tr>
<tr>
<td>Transaction 5</td>
<td>7</td>
<td>4</td>
<td>7</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>7</td>
<td>85</td>
<td>215</td>
<td>19</td>
</tr>
<tr>
<td>Transaction 6</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>7</td>
<td>155</td>
<td>285</td>
<td>19</td>
</tr>
<tr>
<td>Transaction 7</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>7</td>
<td>155</td>
<td>285</td>
<td>19</td>
</tr>
<tr>
<td>Transaction 8</td>
<td>7</td>
<td>4</td>
<td>7</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>7</td>
<td>85</td>
<td>215</td>
<td>19</td>
</tr>
<tr>
<td>Transaction 9</td>
<td>7</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>70</td>
<td>355</td>
<td>9</td>
</tr>
<tr>
<td>Transaction 10</td>
<td>4</td>
<td>4</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>7</td>
<td>85</td>
<td>125</td>
<td>8</td>
</tr>
</tbody>
</table>
Heat map depicting the Risk distribution across various transactions

High, Medium and Low Risk Profile in % terms
The Benefits of Risk Based Testing

The proposed Risk Based Testing approach has the benefits as shown:

1. Running the tests in risk order gives the highest likelihood of discovering defects in severity order.
2. It helps maximize quality in a time and resource constrained situation.
3. Preventive activities can be started immediately as problem areas are discovered early in RBT.
4. It is an effective and scientific approach to identify the testing scope, thereby a better test plan.
5. Helps to determine the test focus areas for the release.
6. It helps reduce customer issues per customer year over year. The RBT methods help to identify the functional areas which are more defect-prone. Hence a test strategy built on RBT should help us reduce the CII inflow.
7. Risks can be continuously monitored to know the status of the product and its quality.
8. Last but not the least, it also enables scaling up the test practices across the organization.

Disadvantages of Risk Based Testing

Although risk-based testing has several benefits, it also includes some disadvantages listed below:

1. Unrecognized risks or risks assessed as too low may cause problems if it becomes a reality.
2. If the risks are described too abstractly, it may be difficult to attach a test to an identified risk.
3. If the risk assessment isn’t diligently done, the accuracy of resulting risk profile would be in question.
Conclusion

Risk based testing is a powerful testing technique that can be applied to modern mobile applications as well as traditional desktop and web applications alike. RBT strategy can be used either to design a manual test plan or even to prioritize which tests must be automated on priority in the order of risk.

RBT helps the testing teams to streamline their testing efforts, which in turn helps in mitigating the risk and minimizing the testing efforts, thus, bringing an objectivity to test designing and test management activities.

The goal of risk-based testing cannot practically guarantee a risk-free product. What we can expect from risk-based testing is to carry out the testing with best practices in risk management to deliver a product that balances risks with quality, features, budget and schedule.