TEST FACTORY MODEL ENSURING EFFICIENT AND OPTIMIZED DELIVERY THROUGH RISK BASED TESTING

13th Annual International
SOFTWARE TESTING CONFERENCE - 2013

Authors:
Kavitha Ayappan, Kavitha.Ayappan@cognizant.com
Praseetha Kesavan, Praseetha.Kesavan@cognizant.com
S, Jayasakthi, Jayasakthi.s2@cognizant.com

Organization:
Cognizant Technology Solution
5/535, Old Mahabalipuram Road,
Okkiyam, Thoraipakkam
Chennai- 600097
Abstract:
In this challenging market all the customers expect high quality product to be delivered on time with low budget. To meet such expectations from the customer end, some organizations tend to rush testing by performing inadequate testing which might hamper quality while increasing long-term support costs. There are also organizations which over-test with high expertise hoping to unearth maximum defect, which may delay software’s delivery to the market thus incurring more cost. In this paper we are attempting to introduce test factory aligned with risk based testing where the resources are divided into different towers with specialization in each tower. Risk based testing further adds advantage to this model by examining the level of the business scenarios and the likelihood of any given defect and the business impact of the defect. Such approach helps organization to plan and deliver the applications / product / Services in an optimal way.

How to strike a balance between reducing overall project costs, efforts and test schedules, while assuring quality with enhanced knowledge in the application:
To achieve this, the organization requires consistent, business-driven approach to decide which functions and requirements are most important to test, and in what order with sufficient knowledge among the resources. Test factory model aligned with risk based testing is once such Optimized model achieved through lean methodology. Test factory model encompasses multiple towers. Each towers has specialized set of domain or testing experts focusing on the individual QA function. And also continuous collaboration in real time with Business and Development teams in this approach speeds up delivery process.

Risk based testing along with the Test factory model add efficiency to the testing approach
Risk Based Testing (RBT) is a type of software testing methodology that optimizes the effort required to test a particular application by accepting certain amount of risk of failure. High level functional requirements are classified into unit level requirement and risk is assessed for each of the unit level requirement by determining weightage for its business critical & failure probability factors. Once risk is identified for unit level requirements, optimal test coverage and test case identification for each of this requirement are calculated by performing risk analysis using its functional complexity and test case priority thus implementing the RBT from the design phase itself.
**Challenges in Software testing**

Testing is always under pressure since testing is the last thing done in a project lifecycle. Hence we need to balance between software development life cycle and testing life cycle. 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 satisfaction within strict budget limits.

For a program of large magnitude and complexity, performing testing 'at the end of the lifecycle' will put entire program at risk. Some of the key the risks are given below:

**High number of defects during functional testing, this leads to**

- The project timelines at risk as it takes more efforts to fix the defects found late in the lifecycle
- The program budget at risk as it is more costlier to fix defects found late in the lifecycle especially when the project is having budget constrain.
- Increases maintenance cost as there are high chances of defect getting leaked to the production

**Failure of previously certified applications results in**

- Not performing the regression testing despite of any component upgrades, could affect the applications that are certified already
- Further increases maintenance cost due to fixing and retesting of failed applications. ( Cost of fixing and retesting – any industry data )
- Unsatisfied business community as the failed applications will not be available in production temporarily

The testing activities will also impede due to 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 on which it will run or even proper test processes.
Not having clarity on how much testing is required for any particular application, and of where the most critical defects might be, organizations are faced with a number of unsavory alternatives.

One is to simply reduce the scope of testing or employ unskilled and new entrants. But without a way to prioritize test needs, this risks missing critical defects and thus impacting the test effectiveness (product quality).

A second is to devote more skilled 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.

Another method that organization generally prefers is test case optimization. This attempt to predict based on the complexity of a unit of code, the likelihood and criticality of the defects in it. Since 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. 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.

The fourth option is automating the testing process, which 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.

Considering all the above said approaches, an effective and efficient approach described in this paper is **Test factory model aligned with risk based Testing**. In the test factory model, the resources are divided into different towers where each tower is specialized in a respective activity. The approach also includes Test early and test often approach to detect defects early in life cycle and certify the components. Risk based testing then further adds the advantage which examines the level of the business scenarios and the likelihood of any given defect and the business impact of the defect. This helps to assure companies to spend their time and money on the most critical areas, and find the more number of defects (and the most critical defects) in the shortest time and at the lowest possible cost.
Some of the benefits reaped out of Test factory model over the traditional model is listed below:

- Optimized use of resources with mixed level of expertise available due to tower based model thus achieving reduction in the cost.
- Tasks divided in to multiple work streams (Tower), resulted in higher productivity.
- Individual leads to monitor each work streams (Tower) thus achieving better efficiency in the delivery.
- Early engagement of all Stake holders giving early feedback.

**Test factory model with RBT – Salient features**

1. **Dedicated team for test design- Design tower**
   - Dedicated team of test analysts working very closely with technical and business analyst teams
   - Focusing on continuous delivery of test design keeping project priorities in mind

2. **Dedicated team to Certify Components**

[Type text]
3. **Component Tower**

- Dedicated team of testers focusing on certifying custom built reusable components
- Focusing on early testing of functionality to ease up functional testing window

![Test Factory model diagram](image)

**Fig 2: Test Factory model**

4. **Continuous Functional testing of applications- Execution Tower**

- Dedicated team of testers focusing only on doing functional testing of applications
- Continued execution with testers picking up unit tested applications from Assembly line

5. **Dedicated team for defect retesting- Hyper care Tower**

- Dedicated team of testers focusing on retesting of defect fixes given by Hyper care team
- Ensuring uninterrupted flow of Functional testing
6. Risk Based Testing (RBT) done along with functional testing

- It is a type of software testing methodology that optimizes the effort required to test a particular application by accepting certain amount of risk of failure.

- Risk Exposure= Failure Probability X Business criticality

![Fig 3: Activities of the Towers in Test factory model](image)

This model operates best with respect to financial aspect as the new entrants can also be employed here in the execution tower since the design tower and component tower has all the expertise designing the test cases which makes it very easy and simple for the new resources to perform the test execution.

RBT done with the Test factory Model

The execution tower will perform the Risk Based Testing (RBT) based on the design created by the design towers which optimizes the effort required to test a particular application by accepting certain amount of risk of failure.

The approach involves the following steps:

- Risk Assessment
- Risk Analysis
- RBT Implementation
- Risk Reassessment and Reporting
High level functional requirements are classified into unit level requirement and risk is assessed for each of the unit level requirement by determining weightage for its business critical & failure probability factors.

**Failure Probability:** A measure that directly affects the reliability and robustness of the application. Failure Probability is calculated as the combined weightage of the following criteria:

- Change Type
- Software Maturity
- Defects rate
- No. of affected screens/entities

**Business Criticality:** A measure that directly affects the success or failure of the business. Business Criticality is calculated as the combined weightage of the following four criteria:

- Type of Process
- Impact of Failure
- Frequency of use
- Number/Significance of affected users
Once risk is identified for unit level requirements, optimal test coverage and test case identification for each of this requirement are calculated by performing risk analysis using its functional complexity and test case priority.

**Testing Level = Risk x Functional Complexity**

**Test case selection = Testing Level x Priority**

**Functional Complexity:** It is the interpretation of the testing difficulty derived from the functionalities of application.

Functional Complexity is calculated as the combined weightage of the following four criteria:

- No. of logical process steps
- Amount of data involved
- Dependencies on other systems
- Calculation level in the process

Testing Level can take values: Full, Partial, Basic and None

**Test Case Level:** Risk Analysis at Test Case Level is performed by assigning priority on the basis of the impact (Business or Functional) if the test case does not meet its objective

Test Case Priority can take the following four values:

- 1 - Critical
- 2 – Major
- 3 - Average
- 4 – Low
**Functional Complexity Grade calculation:**

Select a weightage (Max or Intermediate or Min) for each of the four respective criteria. Each criterion can have weightage as Maximum - 30, Intermediate - 18 and Minimum - 8. (Note: The weightages are customizable and as per the Quality Center Standards).

- Calculate their sum.
- Appropriate grade is assigned based on the range of the sum.

**Test Level Calculation:**

Based on the Risk calculated in the Risk Assessment and Functional Complexity grade, appropriate Test Level is assigned as defined in the Test Level Calculation table.

**RBT Test Cases Selection:**

The test cases that are required will be selected based on ‘Y’ criteria as depicted in the below matrix. (Ranges are as per the Quality centre standard)

<table>
<thead>
<tr>
<th>Risk</th>
<th>Functional Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>A – High</td>
<td>1 – High Full</td>
</tr>
<tr>
<td></td>
<td>2 – Medium Full</td>
</tr>
<tr>
<td></td>
<td>3 – Low Full</td>
</tr>
<tr>
<td>B – Medium</td>
<td>1 – High Partial</td>
</tr>
<tr>
<td></td>
<td>2 – Medium Partial</td>
</tr>
<tr>
<td></td>
<td>3 – Low Partial</td>
</tr>
<tr>
<td>C – Low</td>
<td>1 – High Basic</td>
</tr>
<tr>
<td></td>
<td>2 – Medium Basic</td>
</tr>
<tr>
<td></td>
<td>3 – Low Basic</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Functional Complexity Grades</th>
<th>Ranges</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 – High</td>
<td>76 – 120</td>
</tr>
<tr>
<td>2 – Medium</td>
<td>52 – 75</td>
</tr>
<tr>
<td>3 – Low</td>
<td>32 – 51</td>
</tr>
</tbody>
</table>

**Table 1: Functional Complexity Grade calculation and Test level calculation**
<table>
<thead>
<tr>
<th>Test Case Priority</th>
<th>Requirement Analysis Results</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1 - Full</td>
</tr>
<tr>
<td>1 - Critical</td>
<td>Y</td>
</tr>
<tr>
<td>2 - High</td>
<td>Y</td>
</tr>
<tr>
<td>3 - Average</td>
<td>Y</td>
</tr>
<tr>
<td>4 - Low</td>
<td>Y</td>
</tr>
</tbody>
</table>

Table 2 : RBT Test Cases Selection table

Risk Re-Assessment & Reporting

Hyper care team will perform the Risk re-assessment based on defects trend found in one round (Full coverage) of testing, Failure probability will be revisited

![Defect retesting by Hyper care Team](attachment:image.png)

Conclusion:

To compete, enterprises must cut testing costs as far as possible, without sacrificing quality of application or delaying critical application launch time. 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. Risk-based testing aligned with test factory model 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 with optimum resources usage. The same has been implemented in one of the Insurance projects and some of the benefits reaped are highlighted below:

**Independent Test Factory**

- Providing unbiased view of Software Quality
- Enforcing entry and exit criteria at every stage of testing life cycle from each tower.
- Testing governance ensuring adherence to Testing objectives by the tower leads.

**Synergies with other project stakeholders**

- Close proximity project stakeholders for real time collaboration
- Significant effort reduction in understanding and analyzing requirements due to knowledge already gained from different towers (specialized in respective activity-like Design, component, hyper care, Execution)

**Limitations on Test Factory Model**

- Factory model will work only with defined entry and exit criteria
- Factory will be more effective if both Development & Testing are in same organization as there will lot of coordination between the towers

**Benefits of RBT aligned Test Factory model**

- Increase in SME in short span of time due to specialized tower based implementation
- Information architecture governing all the applications implemented resulted in better client feedback.
- Better usage of Resource utilization – Flux team in Execution tower
- Smooth running of factory ensures efficiency, time and cost savings at an overall program level After implementing the model there has been an estimated cost saving of 50% owing to reuse of the test design artifacts. Also an expected 40% reduction in execution cost is also realized without automation of test build. This also Reduces the UAT phase saving on delivery timelines.
- Reduction in product lifecycle development time, Reduction in product defects, Improved customer satisfaction
- Continuous collaboration in real time with Business and Development teams to speed up delivery process
• Faster time to market as the Dedicated team of testers focusing only on doing functional testing of applications.
• Early detection of defects and optimized Testing efforts as the Dedicated team of testers focusing on certifying custom built reusable components.
• Better collaboration and faster decision making as the Dedicated team of test analysts working very closely with technical and business analyst teams.
• Faster resolution of defects as the Dedicated team of testers focusing on retesting of defect fixes given by Hyper care team.
• Ensures high levels of repeatability, predictability and test coverage.
• Effective usage of tools enabling complete traceability from requirements to test cases and defects during various phases of the project.
• Accelerated test automation helping in reduction of cycle time and execution costs.
• Providing platform to have more frequent releases annually – This will benefit the business end users in moving away from the erstwhile lower frequency. Which meant having to wait longer for the release of the important Business/IT changes.

References:

• Software Testing in the Cloud By: Scott Tilley; Tauhida Parveen

• Risk Based E-Business Testing (Artech House Computer Library,) Hardcover by Paul Gerrard (Author), Neil Thompson (Author)

• Software Testing by Brian Hambling; Angelina Samaroo


• Internal portal for innovation
**Author’s biography:**

**Kavitha Ayappan** has an overall experience of 11 years with 8 years of experience in Delivery assurance currently employed with Cognizant technology Solutions. She has completed her Masters in Engineering from Madras university. She holds six sigma black belt from MSME and have implemented the same in projects. In addition to that she has obtained certification in ITIL and ISTQB and currently takes care of the Delivery assurance in testing projects.

**Praseetha Kesavan** has an Overall experience of 8 years of Delivery assurance currently employed with Cognizant technology Solutions. She has completed her Master in Application from Madurai. she has obtained certification in CSQA and ISTQB and currently takes care of the Delivery assurance in Testing projects.

**S Jayasakthi** has an Overall experience of 5.5 years of Delivery assurance currently employed with Cognizant technology Solutions for the past 1.5 years. She has completed her Bachelors in Computer Technology. She also has certification in ISTQB. She is currently taking care of the delivery assurance in Testing Projects.

**Appendix :**

**Acronyms :**

**RBT** : Risk Based Testing

**Dev QA** : Development Quality Assurance

**UAT** : User Acceptance testing

**QA** : Quality Assurance

**Hyper care team:** This is the team which is responsible for Defect retesting. They retest the defects identified during UAT as well. They communicate the test status to application testing team