# Estimator Metrics

## Assessment of QA Time & Resources

*This paper enlightens the novel approach for estimation of QA time and resources. Numerous approaches e.g. Function Points Analysis, Feature points, Use Case points or Lines of Code (LOC) available in market specifically for project estimation but QA time and effort estimation generally done through past projects experience or simply through wild guess.*

*Some time it gives correct estimation but the majority of time it confers lack of time for QA activities.*

*This paper is an attempt to offer tiny step of concrete approach for QA estimation using priority of influence elements and impact factor.*

Author:

Amit Bhardwaj| Lead, Quality Engineering
Tekmindz India Pvt. Ltd, Noida (India)
amitb.qa@gmail.com

STC
NOV '09

## Abstract:

### Introduction:

Software QA estimation is a complex, high-risk human endeavor. As such, it's important to combine good estimation techniques with an understanding of the factors that can influence effort, time, dependencies, and resources. Some of these factors can act to slow down or speed up the schedule, while others, when present, can only slow things down.

This paper is an attempt to offer concrete approach "Estimator Metrics" for estimation using priority of influence elements, impact factor, degree and priorities for QA activities.

### Audience:

Project Managers, QA Managers, QA Leads

### Area of Application:

QA Testing Time & Resource Estimation for any software project development.

### Issues and Challenges:

Available techniques in market results correct estimation but most of the time they confers lacks of time for QA activities.

QA time and effort estimations are generally done through past projects experience or simply through wild guess. Nobody follows any streamlined process for the same.

### Benefits:

Our approach would help in appropriating time estimation for QA activities and would also assist in getting a better idea about the constituent of estimation and influence elements involved in projects.

# Table of Contents

## 1.0 Introduction:

**H**ow much time will you require to complete testing activities? It is a complicated query faced by Test Professionals and we often struggle to answer, and, when we do, the response is often, "It is dependable..!

Ideally complete testing must mean that by the time testing is done; the tested system is devoid of all unknown bugs. If you test completely, then at the end of testing, there cannot be any undiscovered errors. Basically, when the matrices involved (those are bug detection and bug fixing) reach a certain level i.e. bug detection decreases and bug fixing increases and certain percentage of the test case is passed, we can assume that testing has been completed.

It can only be achieved through effective test estimation process. This paper is an attempt to offer concrete approach for estimation using priority of influence elements, impact factor, degree and priorities for QA activities

## 2.0 Content

### What is the best approach to Software Test Estimation?

**T**here is no simple answer for this. The 'best approach' is highly dependent on the particular organization, project, the experience and attitude of the personnel involved.
For example, given two software projects of similar complexity and size, the appropriate test effort for one project might be very large if it was for highly secure banking application, but might be much smaller for the other project if it was for a low-cost e-learning application. Test estimation approach that only considered size and complexity might be appropriate for one project but not for the other. An estimate serves as a master plan for a software project covering all aspects like costing, staffing, timing. Hence basing this on pure guess work is a definite NO!

After doing extensive search and discussion with experts, we realized that, on average 60% of the hours spent on the total development project is reserved for functional design, technical design and realization and rest **40% is reserved for testing.****

Testing time can be divided into different section for several QA activities. But it would be more appropriate to understand the estimated complexity, size and risk involved for meticulous QA task before zeroing on test estimation.

Following are the steps to achieve reliable effort estimation:

1. **Constituent of Test Estimation**
2. **Calculation Mechanism for Impact Factor**
    - **A.** Classification of Impact element
    - **B.** Degree of Impact Element
    - **C.** Risk Priority of Impact element
    - **D.** Resultant Factor Representation
    - **E.** Metrics Based on "Degree. of Impact element" & "Risk Priority"
    - **F.** Calculation for Impact Factor
3. **Assumption for Degree of QA activities, Risk Priorities & Resultant Factor**
4. **Identification of QA activities**
5. **Preface of Estimator Metrics**

## Step 1: Constituent of QA Estimation

First of all we need to identify the constituent for estimation which has extreme tie with estimation activities. After doing enormous analysis on the same, we realized seven major constituents that reflect on estimation.
All of these constituent should be defined appropriately that need to be included in approach:

1. ***Approach:*** Before moving forward you must need to know what would be the approach to test (QA activities that you need to tackle…) and how thorough you need to test it. Does it require scenario based testing or Use Case based testing?

2. ***Scope:*** Moving forward you need to know the 'scope' and this does not only include the "size" of the systems we need to test, but we also need to take into account all the possible requirements. What all are the in-scope / out-scope elements.

3. ***Prerequisite Info (Project documentation):*** One of the most important pre-request for QA estimation is availability of appropriate functional / requirement documentation. This is one of the most vital points; you should know about how will they develop the system? (If there is lots of module) What will be the basis for QA activities to be performed? Is the system well documented, or there is lack of documentations. A reliable estimation strictly depends upon keen study of project documentation.

4. ***H/W & S/W availability:*** This is something you need to know before giving any estimate. What type of H/W and S/W resource required for the project. Do we need to buy some specific tool or server for the same? It would answer "How Fast" can we test?

5. ***Man Power Efficiency:*** How experienced and efficient is your team to execute QA activities or do we need to hire more skilled people. It is more important incase when automation and performance effort is required or application has some specific areas those require domain expertise.

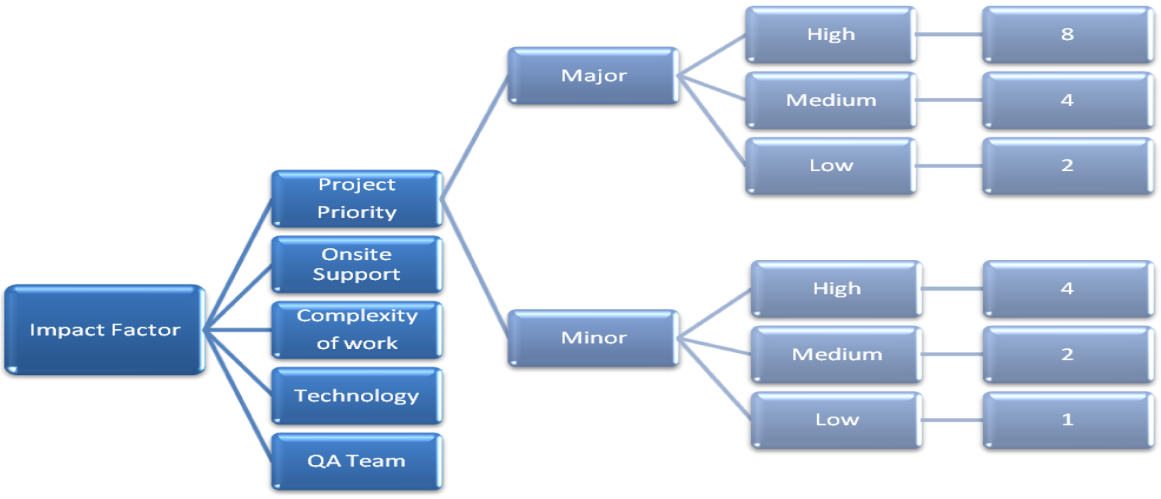
6. ***Risk & Mitigation Plan:*** Before designing estimation you need to discover possible latent risk involved and be ready with the mitigation plan accordingly. It is something needs to be conveyed in estimation document. Lack of documentation, resources (HW, SW and people), Domain knowledge.

7. ***Client's zeal for Quality:*** It is the most important fact for quality output. As a matter of fact, QA testing is not couple of night job rather then it's a long lasting process for quality improvement. Full-flagged QA cycle require to be scheduled in project plan and we need to decide "when to stop testing"

All of these constituent (elements) should be kept in mind before proceeding to estimation. Apart from these elements, the most important dependability factor is the eagerness for quality among entire of the project team. Because it is one of the most important asset for the entire task that would be carried out in future.

## Step 2: Calculation Mechanism for Impact factor

**N**ext stair for estimation is to compute impact factor for concerning project. Meanwhile in order to calculate impact factor we need to know exact elements which have major impact on the estimation task. These elements would be standard for our approach to calculate Impact factor. Constituent of estimation should be kept in mind while you design estimation & impact element need to be defined accordingly.




### A. Classification of Impact Element:

1. **Project Priority:** What is the business priority of your project? What is the impact on the business process? (Human Resource Management, Inventory Control System, Accounting & Budgeting and eLearning Solutions)

2. **Onsite Support:** Full time Onsite QA engineer / Lead will be available or not. It's essential for better communication between dev and test team. (Specifically when client situated at remote location). It can be omitted incase of in-house development.

3. **QA Team:** How proficient is QA team? Do they have appropriate skill set and test ware for the proposed task.

4. **Technology:** Are we using proven technology or is it something new (even experimental)? Wheatear application is a web based application or window based application. It is very important incase client is also anticipating project automation and performance testing.

5. **Complexity of work:** The degree of complexity within the application we need to test as well as the interfacing with other applications. Wheatear any third party interaction is available like online payment gateways or data processing on historic data received from external web application.

### B. Degree of Impact element:

Now we have defined the impact elements, furthermore we need to identify the valid degree as per business impact for all the impact elements. We can characterize two type of degree for subsequent impact element i.e.

- **Major & Minor** (Major implies the soaring effort while Minor implies fewer effort )

| Degree of Impact element | | |
|---|---|---|
| 1 | Project Priority | Major / Minor |
| 2 | Onsite Support | Major / Minor |
| 3 | QA Team | Major / Minor |
| 4 | Technology | Major / Minor |
| 5 | Complexity of work | Major / Minor |

Table: Assumption for degree of impact element

### C. Risk Priority of Impact element:

Next step is to define the business priority of risk involved for impact elements. Normally three type of business priority are considered for the same.

1. **High Risk:** Would require high effort / time to handle the impact element.
2. **Medium Risk:** Would require average effort / time to handle the impact element.
3. **Low Risk:** Discipline within the project is familiar with impact element.

Now each of the impact elements has been described with respect to priority of risk.

### D. Resultant Factor representation

(A numeric representation to calculate Impact Element)

It is the turn to assign some numeric representation for risk priorities. We called it as Resultant Factor and it is merely a numeric representation for Risk Priority defined in previous step. We are assuming low resultant factor for Minor degree as '1', which is then subsequently increasing for the respective combinations.

| Degree of Impact element | Risk Priority | Resultant Factor |
|---|---|---|
| Major | High | 8 |
| | Medium | 4 |
| | Low | 2 |
| Minor | High | 4 |
| | Medium | 2 |
| | Low | 1 |

Table: Assumption for resultant factors

## E. Metrics based on Degree of Impact element, Risk Priority & Resultant Factor

Below table demonstrates resultant factor for all the impact element corresponding to measured (for hypothetical project) degree of impact elements.

| Impact elements | Degree of Impact element | Risk Priority | Resultant Factor |
|---|---|---|---|
| Project Priority | Major | High | 8 |
| | | Medium | 4 |
| | | Low | 2 |
| Onsite Support | Minor | High | 4 |
| | | Medium | 2 |
| | | Low | 1 |
| QA Team | Major | High | 8 |
| | | Medium | 4 |
| | | Low | 2 |
| Technology | Minor | High | 4 |
| | | Medium | 2 |
| | | Low | 1 |
| Complexity | Major | High | 8 |
| | | Medium | 4 |
| | | Low | 2 |

Table: Resultant factors Metrics

## F. Calculation of Impact Factor

Now we have enough information to calculate the value of Impact factor. We would have to consider effective degree of impact element, risk priority and resultant factor applicable to concerning project.
We need to add applicable resultant factor values and divide the sum by number of Impact element defined in step 1, i.e. 5.

**Impact Factor = [Σ Resultant Factors / No. impact element]**

It can be understood by following table where we have assumed fictional degree of impact element.

| Impact elements | Degree of impact element | Risk Priority | Resultant Factor |
|---|---|---|---|
| 1. Project Priority | Major | High | 8 |
| 2. Onsite support | Minor | Medium | 2 |
| 3. QA Team | Major | High | 8 |
| 4. Technology | Minor | Low | 1 |
| 5. Complexity | Major | Medium | 4 |
| Σ Resultant Factors | | | 23 |
| Impact factor( Σ Factors / No. impact element) | | | 4.60 |

Table: Impact factor Metrics

Impact factor is calculated as (=23 / 5) **4.60**

The journey till calculation of Impact factor is very much common and lots of approaches attempt to apply such concept.

Going forward we would define the QA activities and their degrees, risk priorities & resultant factors, not for calculation of any further impact factor but it would help us to priorities our QA activities. Using these element and Impact factor that was calculated in prior calculation (step1-2) would generate actual time and effort estimate.

## Step 3: Assumption for Deg. of QA activities, Risk Priorities & Resultant Factor

(A numeric representation to calculate Test time Estimation using "Estimator Metrics")

We need to define degree of QA activities and corresponding resultant factors.

| Degree of QA Activities | Risk Priorities | Resultant Factor |
|---|---|---|
| Critical | High | 32 |
| | Medium | 16 |
| | Low | 8 |
| Major | High | 16 |
| | Medium | 8 |
| | Low | 4 |
| Minor | High | 8 |
| | Medium | 4 |
| | Low | 2 |

Table: Assumption for QA activates priority and Resultant factor.

## Step 4: Identification of QA activities

"Quality" refers to all the good things that we would like to see in our product. We build a quality product and assure its quality by keeping quality in mind all the time by performing different activities on different levels. Core QA activities can be defined as below:

- RBD / SRS / Use Case documents review
- Test Case Designing
- Test Case Updation
- Functional QA (Test Case execution)
- Performance QA
- Functional Automation

More QA activities can be defined as per requirement. Available features and functionalities can be break downed for all the respective QA activities.

## Step 5: Preface of Estimator Metrics

**B**elow is the preface of 'Estimator Metrics' used to calculate test time for identified QA activities. We have already declared degree and resultant factors for QA activities. Application features can be traced down with all the QA activities. Risk Priorities can be assigned to individual feature for respective Degree of QA activity. A numeric representation for applicable risk priority can be assigned to the same.

| QA Activity | Degree: QA Activity | Feature/ Task | Risk Priority | Resultant Factor | Impact Factor | Avg. Hour = Risk Indicator * Resultant factor | No. of Days per Person |
|---|---|---|---|---|---|---|---|
| **One time activity** | | | | | | | |
| Requirements doc Review / **Use** Case Review. | Minor | F1 | High | 8 | 4.6 | 36.8 | 4.6 |
| | | F2 | Medium | 4 | | 18.4 | 2.3 |
| | | F3 | Low | 2 | | 9.2 | 1.15 |
| Test Case Designing | Major | F1 | High | 16 | | 73.6 | 9.2 |
| | | F2 | Medium | 8 | | 36.8 | 4.6 |
| | | F3 | Low | 4 | | 18.4 | 2.3 |
| **Recurring Activity** | | | | | | | |
| Functional Testing (Application QA) | Major | F1 | High | 16 | 4.6 | 73.6 | 9.2 |
| | | F2 | Medium | 8 | | 36.8 | 4.6 |
| | | F3 | Low | 4 | | 18.4 | 2.3 |
| Test Case Updates | Major | F1 | High | 16 | | 73.6 | 9.2 |
| | | F2 | Medium | 8 | | 36.8 | 4.6 |
| | | F3 | Low | 4 | | 18.4 | 2.3 |
| Bug Regression / QA Build | Major | F1 | High | 16 | | 73.6 | 9.2 |
| | | F2 | Medium | 8 | | 36.8 | 4.6 |
| | | F3 | Low | 4 | | 18.4 | 2.3 |
| **Sub total for One time activity** | | | | | | **193.2** | **24.15** |
| **Sub total for Recurring activity** | | | | | | **374.6** | **48.3** |
| **Sub Total / QA Cycle (Man Hours)** | | | | | | **567.8** | |
| **Sub Total (Man Days)** | | | | | | | **72.45** |
| **QA Estimation = (One time activity+3* Recurring activities)** *(Assuming 3 bug fix builds) | | | | | | | **169.05** |

## Case Study: Implementation of estimator on actual QA project (eInsurance)

| QA Activity | Degree of QA Activity | Feature/Task | Risk Priority | Resultant Factor | Impact Factor | Avg. Man Hour = Impact Factor * Resultant Factor | No. of Days / Person |
|---|---|---|---|---|---|---|---|
| One Time QA Activities | | | | | | | |
| Requirements / SRS Review | Minor | MOTOR POLICY | | | 4 | | |
| | | 1. Policy Registration | Medium | 4 | | 16 | 2 |
| | | 2. Policy Renewal | Medium | 4 | | 16 | 2 |
| | | 3.Claims Management | High | 8 | | 32 | 4 |
| | | ACCIDENT POLICY | | | | | |
| | | 1. Policy Registration | High | 8 | | 32 | 4 |
| | | 2. Policy Renewal | Medium | 4 | | 16 | 2 |
| | | 3.Claims Management | High | 4 | | 16 | 2 |
| | | BOND POLICY | | | | | |
| | | 1. Policy Registration | Medium | 4 | | 16 | 2 |
| | | 2. Policy Renewal | Medium | 4 | | 16 | 2 |
| | | Fire/Extraneous Perils Policy | | | | | |
| | | 1. Policy Registration | Medium | 4 | | 16 | 2 |
| | | 2. Policy Renewal | Medium | 4 | | 16 | 2 |
| | | 3.Claims Management | High | 8 | | 32 | 4 |
| | | Marine Policy | | | | | |
| | | 1. Policy Registration | Medium | 4 | | 16 | 2 |
| | | 2. Policy Renewal | Medium | 4 | | 16 | 2 |
| | | 3.Claims Management | High | 8 | | 32 | 4 |
| | | Aviation Policy | | | | | |
| | | 1. Policy Registration | High | 8 | | 32 | 4 |
| | | 2. Policy Renewal | Medium | 4 | | 16 | 2 |
| | | 3.Claims Management | High | 8 | | 32 | 4 |
| | | | | | | 368 | 46 |
| Test Case Designing | Major | MOTOR POLICY | | | 4 | | |
| | | 1. Policy Registration | Medium | 8 | | 32 | 4 |
| | | 2. Policy Renewal | Medium | 8 | | 32 | 4 |
| | | 3.Claims Management | Medium | 8 | | 32 | 4 |
| | | ACCIDENT POLICY | | | | | |
| | | 1. Policy Registration | High | 16 | | 64 | 8 |
| | | 2. Policy Renewal | Medium | 8 | | 32 | 4 |
| | | 3.Claims Management | Medium | 8 | | 32 | 4 |
| | | BOND POLICY | | | | | |
| | | 1. Policy Registration | Medium | 8 | | 32 | 4 |
| | | 2. Policy Renewal | Medium | 8 | | 32 | 4 |
| | | Fire/Extraneous Perils Policy | | | | | |
| | | 1. Policy Registration | Medium | 8 | | 32 | 4 |
| | | 2. Policy Renewal | Medium | 8 | | 32 | 4 |
| | | 3.Claims Management | Medium | 8 | | 32 | 4 |
| | | Marine Policy | | | | | |
| | | 1. Policy Registration | Medium | 8 | | 32 | 4 |
| | | 2. Policy Renewal | Medium | 8 | | 32 | 4 |
| | | 3.Claims Management | Medium | 8 | | 32 | 4 |
| | | Aviation Policy | | | | | |
| | | 1. Policy Registration | High | 16 | | 64 | 8 |
| | | 2. Policy Renewal | Medium | 8 | | 32 | 4 |
| | | 3.Claims Management | Medium | 8 | | 32 | 4 |
| Sub Total of the required Man Hr. and Man Days for QA activities | | | | | | 608 | 76 |
| Total time (Man Hr.) required for QA Activities | | | | | | 976 | |
| Total Man Days required for QA Activities (Total Man Hrs. / 8*) | | | | | | | 122 |
| Total Man Days required for 2 QA Resources (Total days/ 2) | | | | | | 61 | |

*1 Man Day = 8 Man Hrs.

## Case Study: Actual "Time taken with extended hours" for two QA task

| Modules/Feature | SRS/BRS Review | Test case Designing | Subtotal Hr. |
|---|---|---|---|
| **MOTOR POLICY** | | | |
| 1. Policy Registration | 10 | 30 | 40 |
| 2. Policy Renewal | 10 | 24 | 34 |
| 3.Claims Management | 24 | 24 | 48 |
| **ACCIDENT POLICY** | | | |
| 1. Policy Registration | 24 | 40 | 64 |
| 2. Policy Renewal | 12 | 24 | 36 |
| 3.Claims Management | 12 | 24 | 36 |
| **BOND POLICY** | | | |
| 1. Policy Registration | 12 | 24 | 36 |
| 2. Policy Renewal | 12 | 24 | 36 |
| **Fire/Extraneous Perils Policy** | | | |
| 1. Policy Registration | 16 | 24 | 40 |
| 2. Policy Renewal | 12 | 24 | 36 |
| 3.Claims Management | 12 | 24 | 36 |
| **Marine Policy** | | | |
| 1. Policy Registration | 10 | 24 | 34 |
| 2. Policy Renewal | 10 | 24 | 34 |
| 3.Claims Management | 24 | 24 | 48 |
| **Aviation Policy** | | | |
| 1. Policy Registration | 24 | 42 | 66 |
| 2. Policy Renewal | 10 | 24 | 34 |
| 3.Claims Management | 24 | 24 | 48 |
| **TOTAL time (Man Hrs)** | **258** | **448** | **706** |
| **Total Man Days** | **33** | **56** | **87** |

| | | | |
|---|---|---|---|
| **Extra time buy from Client** | | | **30** |
| **Actual "Time taken with extended hours" for two QA activities** | | | **117** |
| **Estimator time for same** | | | **122** |

Due to earlier Inadequate Estimation, we forced to buy extra time from client.
First Estimation time = 87 days
Forced to buy extra time = 30 days
Total time taken to finish the tasks = 117
QA time estimation for the QA activities using Estimator = 122

**Estimator approach seems much better than earlier one!!!**

## 3.0 Conclusion

This is our endeavor to present a concrete approach for QA time estimation; we have taken some assumptions to calculate the impact factor for the risk involved in a project and influence element of the project to make an efficient time estimate for QA activities.

We have implemented this approach on couple of projects successfully and the time calculated through Estimator metrics was almost accurate. When an approach is being followed, than it becomes ideal and we believe the day is no longer far behind, when we will see such approach would be followed globally by all the software leaders

Using this approach, we are sure that one can not only get appropriate time estimation for QA activity but also gets a better idea about the constituent of estimation and influence elements involved in the projects.

The main idea behind sharing this approach in the public is to make, QA world aware about the need of some concrete approach to estimate the time for QA activities. So that they can avail the benefits of this approach, also this approach could be a stepping stone as more people might come out with more ideas to further refine this approach, comprising it into a more reliable and efficient estimation technique.

## 4.0 Definitions, Abbreviation and Acronyms

| Acronym | Description |
|---|---|
| **Constituent of QA Estimation** | Seven major elements that reflect on estimation |
| **Impact Element** | Five slandered element are considered for Impact Factor calculation |
| **Degree of Impact element** | Two type of degree is defined i.e. Major & Minor (Major implies the soaring effort while Minor implies fewer effort ) |
| **Risk Priority of Impact element** | Three type of Risk Priority is defined i.e. High, Medium and Low for impact elements. |
| **Resultant Factor** | A numeric representation for Risk Priorities |
| **Impact Factor** | Calculated factor, Impact Factor = [Σ Resultant Factors / Σ No. of impact element] |

## 5.0 References


| Item | Description |
|---|---|
| **QA Estimation: Robin Thomas** | http://www.testinglounge.com/2007/06/12/qa-estimation |
| **QA Time Estimation: Krishantha** | http://softwareqaq.blogspot.com/search/label/QA Time Estimation |
| **The testing estimation process: Antonio Cardoso** | http://sqa.fyicenter.com/art/The_testing_estimation_20process.html |


## 6.0 Acknowledgements


| Name | Description |
|---|---|
| **Satish Kumar** | Satish, he is a QA Manager and Co-author in this paper. We had long discussions on the various aspects for QA estimations. |
| **Avnish Pundir** | Avnish, he is a Technical Manager and helped us to understand various technical aspects related to project estimation. We are truly thankful to him for his extended support and guidance. |
| **Prachi Sharma** | Prachi, she is head of organization and always takes keen interest to offer extensive support for those who want to expand technical horizon. She provided us full support and guidance during our paper preparation work. |
| **Animesh Matiman** | Animesh, he is a Technical Manager and we had long brain storming sessions to include quantitative approaches during the estimation process. We are truly thankful to him for his extended support and guidance. |

## Biography of the Author

### Meet Amit Bhardwaj

Amit Bhardwaj, Lead in Quality Engineering at TekMindz India has over six years of professional experience in Quality Assurance and software testing. He is accountable for the responsibilities of QA test cycle, defect management, creating Metrics and asset creation (tools and processes).

Amit holds Masters Degree in Computer Science from GBTU, Lucknow. He is widely involved in the testing presentations and paper submission.

He had been selected among seven finalists for the Accenture software testing challenge 2007, Bangalore, where he presented his solution based on case study comprising Test Plan, Strategies & Test Estimation.

Prior to joining TekMindz he worked at QA InfoTech for almost three year. He has also imparted paper presentation on "Testing: The Magnitude of three dimension skill set" to trainee' batches @ QAIT. He has also played a lead role in providing knowledge transfer sessions to new joiners and other employees on contemporary technologies such as Cloud Computing and Selenium.

He is an extracurricular guy. He takes keen interest in reading, playing Table Tennis, Snooker and Computer Games.