



RDCT Software Experts for Healthy Software Products

¹N Prameela, ²Raghu Dhumpati

¹Department of Computer Science and Engineering, Sir Vasavi College of Engineering, Tadepalligudem,

²Department of Computer Science and Engineering, Dr. Paul Raj Engineering College, Bhadrachalm

Email: ¹prameelakotipalli@gmail.com, ²raghu.dhumpati@gmail.com

Abstract— Technology is growing rapidly in today’s world. The system that produces technology is also changing. According to surveys it tell that unemployment is high and most of the company’s are slashing there expenditure to meet there requirements. Companies are not maintaining good employees. When no quality employees then no quality product. These terms are interrelated to each other. In this period no company can offer good quality product. Human inventions are changing based on towards world requirements but still software industry is facing problems for its survival. Experienced computer professionals and upcoming computer professional are victims of present precision. Software industry is now a patient on ventilator. Most of software companies are winding up because of precision and some exist with limited employees. Our paper gives explanation about how a software industry can maintain quality within short span. The cost of software problems or errors is a significant problem to global industry, not only to the producers of the software but also to their customers and end users of the software. There is a cost associated with the quality of software to companies who purchase a software product and also to the companies who produce the same piece of software. The task of improving quality on a limited cost and time base is a difficult one.

The foundation of this paper lies with the difficult task of evaluating software from its inception through its development until its testing and subsequent release. Even we can make software project success with few employees in the company. The focus of this paper is on the improvement of the testing of software quality with a limited budget and reduced cost of time.

Index Terms— Software Problems, Software Quality, Technology, Testing, Ventilator.

I. INTRODUCTION

A. Present situation of Software Employees

Japan like country is booming in humanoid and robotics. But India, USA and European countries which depend on software industry are facing problems. They name it as precision period. In this period they reduced staff/employees because lack of software projects. As days are going on expenditure of companies increasing

and no one shows interest towards software projects. This is main cut-throat to the software industry. So Employees are maintaining consistency in same company. Some software industries are sending employees on long live and some of them even terminating existing employees. The main reasons for software crisis are lack of new software projects. Because of this unemployment in technical field like computer science and engineering and information technology are suffering. Lot of well worth computer and information technology professionals are facing problems. Hiring trends in top Industry sectors is shown in Fig. 1.

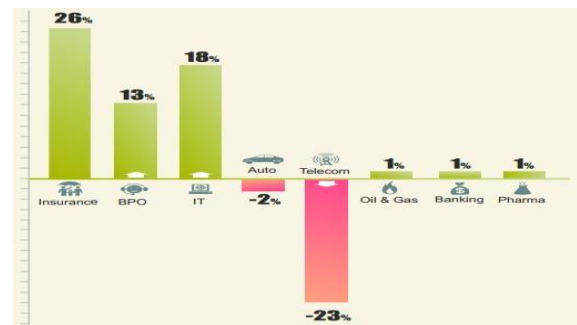


Fig. 1 Hiring trends across top industry sectors (2014).

Based on the current situation employees should work efficiently. They should poses cross platform skills.

B. Present situation of Software Companies

They cannot disappear from global and they pretend that they exist. But there are lots of problems surrounding them as tsunami. One major problem is the financial problem, to face such problems they have to reduce or cut-off expenditure. Reduce employee size, perks, incentives, bonus soon. If we compare recruitment from last 3 to 4 years this year it is incremented from last year (2013). Some fired software employees are committing suicide along with there families. Based on some or other reasons, the software industry facing precision and unable to recruit good software professionals. When no worthy professionals, then they cannot obtain quality. Now non-IT industries revenue is good and more when comparing with IT industry as shown in Fig. 2 and 3. Now

the graph related to software industry is touching base of it. It is our responsibility to rise it up.



Fig. 2 Hiring trends across functional areas.



Fig. 3 Hiring trends across top cities in India (2014).

Based on current situations a company should support employees by adopting them for training under cross platforms. Giving them perks, incentives, bonus, sending for higher studies, for better training and spending more profit on them. Company and Employees are like important for software project.

C. Customer should benefit

According to above said information whatever the problems may surmount Software Industry but customer should be benefited with good quality of software.

D. Competitive Landscape of the industry

Present Software Industries are facing competitiveness in Fig. 4. They want to exhibit that there share or revenue on projects is more. Even some Software Companies are showing fake or false revenue income. There are few things in considering competitiveness in the Software industry. They are given as follows.

1. Financial Problems
2. Employee Treats
3. Treat of new Entrants

4. Rivaistry among Employees
5. Bargaining Power of Buyers
6. Bargaining power of Suppliers
7. Treat of Substitutes

Even there may be problems but the client should be supplied with good quality software or error defect software.

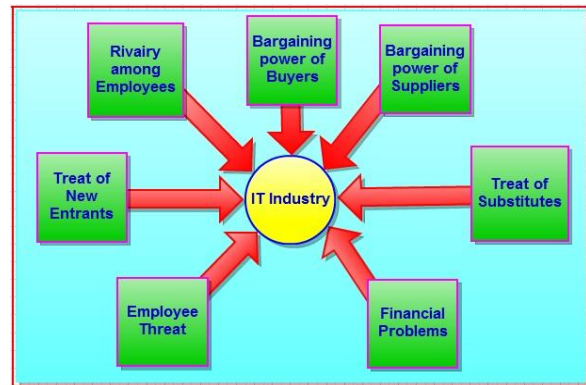


Fig. 4 Competitive Landscape of the Industry.

II. PROBLEMS RELATED SOFTWARE PROCESS

A. Software related problems

Inception of software development life cycle starts at requirements elicitation and analysis, later designing and implementation. This should be done in Fast Track. Model for Fast Track is given Fig. 5. Software related problems should identify and solved as early as possible. In Fast Track model capture information as early as possible by using various techniques. End user should stay with Software Professionals work area for more than 3 days if possible or professionals should travel to client area or video conference. Based on End User requirements make a template. This template can change at any stage. The changes should be like that they should be minor or simple. We cannot measure requirement. The requirement should not be major. Software Professional work should be like that it is easy. We can recruit new employees for effective collection of requirements. We can use existing employees also. After that we carry our work to Design and Implementation phase. All three phase Requirements Elicitation, Analysis and Design or Architecture phase should be completed as early as possible. Software Professionals can use template Requirements Elicitation → Analysis → Design triangle where Software Professional identifies requirements, analysis and Prototype of designs at simultaneously around speculated time. The three phases should be tested whether they meet the goals which we planned. We make rotations till we satisfy with the system.

After completion of three stages we enter into implementation, testing stages. Quality of the software is based on test results. Whether we maintain Software Quality or not depends on test. If testing is done up to User, Client and Stakeholders requirements design, then possibility of Quality.

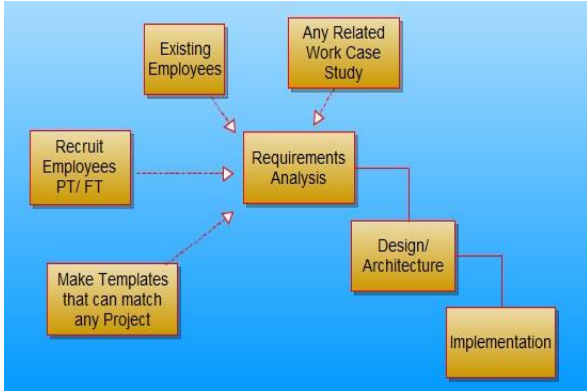


Fig. 5 Fast Track Model

Implementation is nothing but coding. Coding is done as per the prototype screens or model. Next part after implementation is testing, which is trivial.

B. A Test Cycle

Although variations exist between organizations, there is a typical cycle for testing as shown in Fig. 5. The sample below is common among organizations employing the Waterfall development model. The same practices are commonly found in other development models, but might not be as clear or explicit.

- Requirement Elicitation: Testing should begin in the requirements phase of the software development life cycle. During the design phase, testers work to determine what aspects of a design are testable and with what parameters those tests work.
- Testing Plan: Test strategy, test plan, testbed creation. Since many activities will be carried out during testing, a plan is needed.
- Test Deployment: Test procedures, test scenarios, test cases, test datasets, test scripts to use in testing software.
- Test Execution: Testers execute the software based on the plans and test documents then report any errors found to the development team.
- Test Reporting: Once testing is completed, testers generate metrics and make final reports on their test effort and whether or not the software tested is ready for release.

- Test Report Analysis: Or Defect Analysis, is done by the development team usually along with the client, in order to decide what defects should be assigned, fixed, rejected (i.e. found software working properly) or deferred to be dealt with later.
- Defect Retesting: Once a defect has been dealt with by the development team, it is retested by the testing team.
- Regression Testing: It is common to have a small test program built of a subset of tests, for each integration of new, modified, or fixed software, in order to ensure that the latest delivery has not ruined anything, and that the software product as a whole is still working correctly.
- Test Closure: Once the test meets the exit criteria, the activities such as capturing the key outputs, lessons learned, results, logs, documents related to the project are archived and used as a reference for future projects.

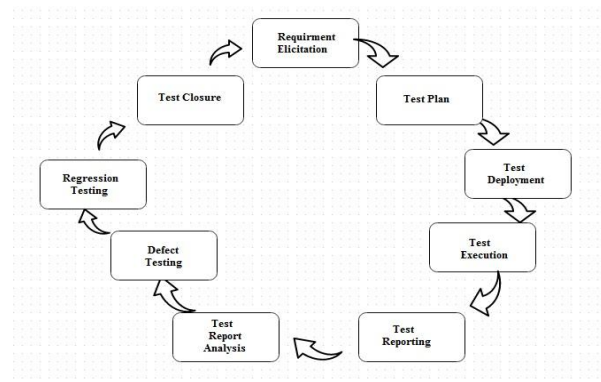


Fig. 5 Test Cycle

This testing cycle is useful for software projects within a system. What about embedded systems like space shuttle, huge machineries, mobiles and any other electric gadgets that contain software.

Testing plays key role in success of a software project. Testing means not debugging but handling Failures, Defects and Mistakes as shown in Fig. 6.

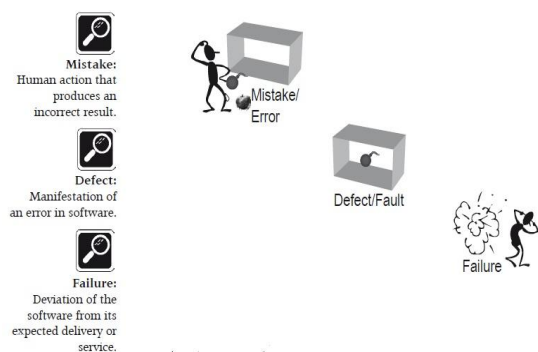


Fig. 6 Handling Failures, Defects and Mistakes.

Process improvement uses analysis of incident reports to find areas where mistakes may be prevented by new ways of working (new processes) or caught earlier by better quality assurance processes as shown in Fig. 7.

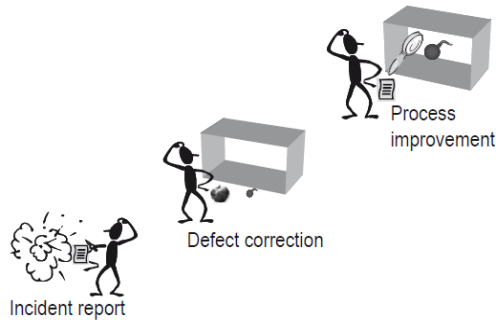


Fig. 7 Process Improvement.

C. Principles for Software Testing

The purpose of software testing is to detect errors in the software. The tester should ideally detect all errors before the software is released to the customer. Full test coverage of a program is impossible. “Proving that a program is fault free is equivalent to the famous halting problem of computer science. The main principle of software testing “is the process of executing a program with the intent of finding errors”. To test the program more thoroughly a tester would need to evaluate the program to detect both types of errors. This principle is thus more detailed to “Test the program to see if it does what it is supposed to and to see if it does what it is not supposed to do”. In order for the tester to find these errors, we will devise a number of tests to execute on the software itself. These tests must be based on prior knowledge of the software.

III. ANALYSIS

Most of Software Projects are failure because of no proper testing. Even though we spell it is trivial, we neglect it. Recent software failure is Singapore Airliner business class ticket sold for economy price. Article is shown in below Fig. 8.

Computer Error Sells 900 Business Class Tickets at Economy Prices

Jo Piazza
Managing Editor
December 4, 2014
Follow



Fig. 6 Computer Error Sells 900 Business Class Tickets at Economy Prices. There are lot of models and test patterns, but we are failing in real life. Some problems are still unidentified till today.

IV. SOLUTION

One of the important facets of software development life cycle is testing. Software testing is an area that is being considered and given utmost importance in the world of fast changing technology. There are various stages at which testing are done to ensure quality of delivery. Result reporting is a mechanism for presenting to the customer, from different angles, the state of the product. Reporting format varies depending on the

- Stage of the testing in the development life cycle
- Audience at each stage
- Transparency involved during testing – white box or black box
- Type of testing involved like Functional, Performance/Load/Stress, Disaster recovery etc.,

Independent System testing takes about 20 to 50% of the development time depending on various combinations of the types of tests conducted on a product. The complex the testing gets more would be the effort required. Of course, the more a product is tested, the better the quality would be. Typically, test result reporting would consist of about 5 to 10% of this effort. Result reporting is very important and even more so when the product has failed testing. There are various aspects that the customer would be interested in knowing about the product line its performance, platform dependence etc., and just not the compliance to functional requirements.

A product could be tested for various aspects some of which are listed below Fig. 7. It makes a lot of difference to handle each of these aspects while reporting the results. Each of them has its own impact on the state of the application. Strengths and limitations of a product are better presented to the customers only if they are handled individually.

Sl. #	Type of Testing	Priority(High/Med/Low)
1.	Functionality	
2.	Regression	
3.	Security	
4.	Performance (Load, Volume, Response, Endurance etc...)	
5.	Scalability/reliability	
6.	Usability/ Graphical User Interface (GUI)	
7.	Conversion/Migration	
8.	Platform	
9.	System Interoperability	
10.	I18N/Localization	
11.	Certification/Compliance	
12.	Disaster Recovery	
13.	Installation/Upgrade	

Fig. 7 various aspects that should be tested.

A. Testing Process for all phases

Testing is a process. Test process is same for any model or phases. Test process is given in Fig. 8.

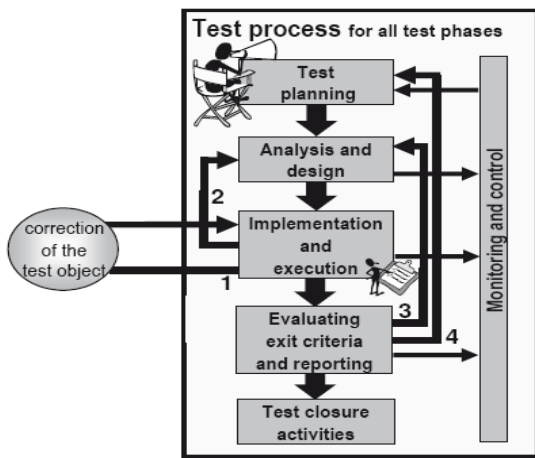


Fig. 8 Test Process.

B. Agile Software Development

Agile software development is a group of software development methods in which requirements and solutions evolve through collaboration between self-organizing, cross-functional teams. It promotes adaptive planning, evolutionary development, early delivery, continuous improvement and encourages rapid and flexible response to change.

C. RDCTSE

We need Requirements Design Coding Testing Software Experts. By having such experts a company can save revenue and produce better software. According Waterfall Model (six phases)

$$\text{Time for Six Phases (T)} = \Sigma(t_1 + t_2 + t_3 + t_4 + t_5 + t_6)$$

$$\text{Total Time} = T$$

$$\text{Teams for Six Phases (T}_T) = \Sigma(T_A + T_B + T_C + T_D + T_E + T_F)$$

$$\text{Total Teams} = T_T$$

$$\text{Amount for Phases} = A$$

$$\text{Total Project} = T + T_T + A$$

RDCTSE we reduce Time for phases, teams and less expenditure. A cross-functional team is a group of people with different functional expertise working toward a common goal. It may include people from finance, marketing, operations, and human resources departments. Typically, it includes employees from all levels of an organization. Members may also

come from outside an organization (in particular, from suppliers, key customers, or consultants). Cross-functional teams often function as self-directed teams assigned to a specific task which calls for the input and expertise of numerous departments. Assigning a task to a team composed of multi-disciplinary individuals increases the level of creativity and out of the box thinking. Each member offers an alternative perspective to the problem and potential solution to the task. In business today, innovation is a leading competitive advantage and cross-functional teams promote innovation through a creative collaboration process. Members of a cross-functional team must be well versed in multi-tasking as they are simultaneously responsible for their cross-functional team duties as well as their normal day-to-day work tasks. Decision making within a team may depend on consensus, but often is led by a manager/coach/team leader. Leadership can be a significant challenge with cross-functional teams. Leaders are charged with the task of directing team members of various disciplines. They must transform different variations of input into one cohesive final output. Cross-functional teams can be likened to the board of directors of a company. A group of qualified individuals of various backgrounds and disciplines are assembled to collaborate in an efficient manner in order to better the organization or solve a problem. Cross-functional team will have a transaction from one phase to next phase. They need not wait for any document from one phase to another. After completion of one phase they get document and its approval for next phases as early as possible.

$$\text{Time for Six Phases (T)} = \Sigma(t_1 + t_2 + t_3 + t_4 + t_5 + t_6) > T_i$$

$$\text{Total Time} = T_i$$

$$\text{Teams for Six Phases (T}_T) = \Sigma(T_A + T_B + T_C + T_D + T_E + T_F) > T_{TS}$$

$$\text{Total Teams} = T_{TS}$$

$$\text{Amount for Phases} = A$$

$$\text{Total Project} = (T_i + T_{TS} + A) < (T + T_T + A)$$

$$\text{Finally Total Project} = T_i + T_{TS} + A$$

Team should participate in almost all phases. It is better to use coding and testing teams combined for better results under con-but situation. In this period a company can easily gain more profit with less amount of time. The normal waterfall model or any other model it takes more time when compared with this agile process.

By doing, team can satisfy client with good quality product. There are three mutually dependent parameters that should be balanced. In fact, for everything we do, we

need to balance these parameters, but here we'll look at them from a tester's view point. The parameters are:

Time: the available calendar time

Resources: the available people and tools

Quality: the quality of the testing

These parameters form what we call the quality triangle as shown in Fig. 9, which is the triangle for the quality of work we can deliver.

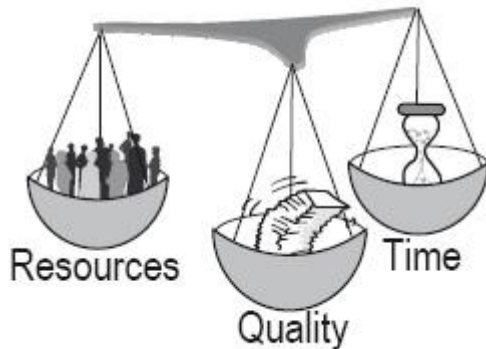


Fig. 9 Quality Triangle.

V. CONCLUSION

Most of the software projects are not success in history. Even tough they are related with cost and time. Our paper will help in better planning of Software, Organizing of Software, Controlling Software and Termination of the Software Project. We are having various tools in for better design, implementation of software and Test Software. It



is mandatory to maintain quality of the Software. We have less number of resources and we have to use it efficiently. For effective use we can hire new employees part-time or full-time, take existing employees, bring User to working area/ contact User by Electronic Gadgets, make templates so that they can be modified at any stage, combine test and quality actives together for better software.

REFERENCES

- [1] "Good Requirements Collection for Better Software Project" By Chodagam Suresh Kumar, Prathipati Ratna Kumar, Dr. G P Saradhi Varma
- [2] "FAST TRACK TECHNIQUE FOR SOFTWARE TESTING AND QUALITY ASSURANCE PRACTICE IN PROJECT DEVELOPMENT LIFE CYCLE" By Chodagam Suresh Kumar, Mikkili Dileep Kumar, Donavalli Venkata Vidya Deepthi, Prathipati Ratna Kumar.
- [3] "SOFTWARE BUG LILLIPUTIANS WHICH CAUSE GIANT DAMAGE TO SYSTEMS" Divya Manisha Seethalam, Valiveti Karthik, Valluru Gowthami, Gudikandula Radha Krishna Murthy, Prathipati Ratna Kumar
- [4] "BASICS FOR EASY CAPTURING AND MODELLING OF REQUIREMENTS ELICITATION, ANALYSIS AND DESIGN IN SOFTWARE DEVELOPMENT PROCESS" By Kadupukotla Satish Kumar, Konda Sreenu, Bathula Prasanna Kumar, Chintagunta Ambedkar.