

# Overcoming Testing Challenges in Project Life Cycle using Risk Based Validation Approach

K. Nageswara Rao

Professor & Head, Department of Computer Science and Engineering, P.V.P.Siddhartha Institute of Technology  
Kanuru, Vijayawada-7, Andhra Pradesh, India.

A. Pathanjali Sastri

Assistant Professor, Department of Computer Applications, V.R.Siddhartha Engineering College,  
Kanuru, Vijayawada-7, Andhra Pradesh, India.

**Abstract** — According to James Whittake, Microsoft Testing Expert and Author, “There are a number of trends that testers are going to have to grapple with. The first is that software is getting better. The result of this is that bugs are going to become harder and harder to find and the weaker testers will be relegated to Darwinian insignificance. Keeping sharp, building skills and maintaining a cutting edge testing knowledge has never been more important”. The key drive for software testing is capturing defects. Testing is becoming a lot more complex. Most software products are not tested thoroughly, so both users and the organizations that write the software expect them to have bugs. There is more and more realization that testing becomes a bottleneck without upfront planning and projects need consider that in software development reducing defects is substantially important. When a project is low on funding or over budget, the first important task that is cut is ‘Testing’. But limiting testing of a solution will increase the chance of system failure or unknown bugs that can cripple your business.

**This paper focuses on the approaches that can be considered to reduce the number of defects captured in testing or later phase of testing.**

**Keywords:** *Domain Test Engineers, Subject Matter Expert, Test Automation, Risk Based Approach.*

## I. INTRODUCTION

The main Challenge for any testing environment will be obviously to test the application for its perfect functionality according to the requirement specifications and well within the acceptable time limit. In well run software organizations testing is not a defect detection activity. Rather, testing should merely verify that the software performs correctly under a wide range of operational conditions. By understanding and addressing the major causes of defects, quality can be designed in from the start, substantially reducing both the cases (a) about 40% of project effort typically spent on rework and (b) the risks to which software exposes business. This paper will cover "best practices" recommendations to help avoid the pitfalls associated with traditional software testing and focus on software testing with the key objectives of reducing the cost of the project test phase. It also provides a framework that assures that reviews are conducted by experienced, qualified, and dispassionate experts and that projects are on track.

This paper is organized as follows. Section II provides basic differences between verification and validation; Section III briefly discusses of importance of software testing in project life cycle; Section IV describes the challenges faced during testing phase of a large project through a case study; and Section V discusses in detail the analyses of the above challenges/problems.

## II. VERIFICATION VS VALIDATION

Verification answers the question, "Are we building the product or system right?" It involves doing the right things up front in a software development project - using best practices for requirements, analysis, design, construction, deployment, and monitoring and ensuring auditable workflows throughout. Verification is the process of doing reviews and walk-through and conducting interviews. Validation answers the question, "Are we building the right product or system?" Many vendors today are building their products right, but studies show that they are not always building the right products. Validation requires involvement from all stakeholders in the specification of requirements and throughout development. Validation is the process of doing actual testing in the source code.

#### A. *Quality Objective – Verification And Validation*

Verification makes sure that the product is designed to deliver all functionality to the customer; it includes reviews and meetings to evaluate documents, plans, code, requirements and specifications [1]; this can be done with checklists, walkthroughs, inspection meetings, etc. validation typically involves actual testing and takes place after verifications are completed, which ensures that the system meets the full requirements [1]. So both verification and validation ensures that at the end of the project development cycle, the user should find that the project has met or exceeded all of their expectations as detailed in the requirements. Any changes, additions, or deletions to the requirements document, Functional Specification, or Design Specification will be documented and tested at the highest level of quality allowed within the time of the project and within the ability of the test team. The objective of verification/validation cycle is to identify and expose all issues and associated risks, communicate all known issues to the project team, and ensure that all issues are addressed in an appropriate matter before release. As an objective, this requires careful and methodical testing of the application to first ensure all areas of the system are scrutinized and, consequently, all issues (bugs) found are dealt with appropriately.

### III. IMPORTANCE OF TESTING IN IT PROJECTS

In some cases an IT organization, inexperienced managers sometimes gamble on the success of a project by skipping thorough testing or having programmers do post-development functional testing of their own work, a decidedly high risk gamble. A test engineer typically needs to have a perspective of what might go wrong with this functionality, and how to ensure that the product meets expectations. It is difficult to get a technical person who can be highly effective in approaching tasks from both of those perspectives. But there is a heavy demand for test specialists with domain knowledge, testing skills and technical skills. If the team has fairly fresh inexperienced staff, and lack of required skilled staff as a part of the test team will definitely hamper the project deliveries. Thus a test team competency plays a vital role in the projects. The competency of the team has its impact on the quality of deliveries based on the expertise knowledge & skills present in the team members. Higher the competency in the team, higher is the efficiency and effectiveness in the deliveries. Testing has to be accepted as vital part of any software development project and sooner or later, organizations need to bring in domain testers (discussed in Section – V. D) to ensure accurate testing that is essential for the software to be implemented successfully.

### IV. CHALLENGES FACED WHILE TESTING IN LARGE PROJECTS – CURRENT SCENARIO

Testing has always been complex and the most challenging part in the project life cycle. Though the timelines for various stages of project are defined during the kick-off phase, most of the times in the real time scenario, testing cycle has always been crunched and shortened. Poor Testing is one of the major factors about the failure of project, and we have to accept that the tester shares not all but maximum accountability for the failure of the project. Finally, if testing team starts testing the product in that given short time period, the testers do not have any other option except to complete the activity like a mechanical process and give green signal to the module that is buggy. Because of delivery urgency the same buggy module goes for live, which will definitely ruin the project and future business of company. So in order to avoid the project failure just because of testing, the testers need to be given enough timeline for testing. The project team should understand that the testing activity has to be carried out successfully not to blame developers for the bugs but to bring the product out very healthy. Apart from this when analysed few failed projects, in most cases it is due to the tester who failed to understand the user requirements and this happened in the following cases:

- The tester interprets the requirements incorrectly and assumes things while testing.
- Testers lack sufficient domain knowledge in the team and are dependent on the developers to understand the requirements and have no contact with the actual users.
- Testers' involvement late in projects and are not involved right from the requirement gathering from client.

The following case study helps us to understand the challenges aroused during the testing phase of a large project in the health care domain.

#### A. *Case study - Healthcare automation project of ABC hospital (Name changed to maintain confidentiality)*

Our study on a healthcare project XYZ (name changed to maintain the confidentiality) posed many challenges for its complexity in design, development and testing the patient data, diagnosis results,

safety characteristics etc. Testing Healthcare products requires more of domain knowledge and testing in general is very critical in every stage of the life cycle of software development activity. On the other hand, Healthcare products has to conform to Safety and Regulatory standards i.e, DICOM, HL7, or HIPAA standards which make good business sense to be in market competing with the other vendors. This in turn makes the test engineers face with multiple challenges while testing the software and delivering to the customer with high quality. Healthcare product testing requires more of domain knowledge, and testing in general is critical in the life cycle stage of any software development activity. In today's situation, healthcare software is gaining more importance and is becoming more complex with the incorporation of new features and design based on customer feedback and market needs. The following are some top software testing challenges we faced during the project. Some of the challenges/risks faced by testers while testing in the absence of skilled testers, lack of sufficient healthcare domain knowledge, not involving test team during the development life cycle etc. are listed below.

TABLE I. SOFTWARE TESTING CHALLENGES FACED WHILE DEVELOPING AN HEALTHCARE PRODUCT

S.No	Challenges Faced	Results
1	Lack of Skilled testers	Resulted into incomplete, insufficient and inadequacy of testing that led to spending of lot of effort in finding and reporting the bugs.
2	Lack of availability of standard clinical test data / datasets during testing	Lead to insufficient test coverage.
3	The team members had insufficient knowledge of Healthcare domain standards	Resulted in inadequate testing.
4	Poor understanding of requirements and Miscommunication or no communication with the end-users during testing/development cycles	No specifics of what an application should or shouldn't do (the application's requirements) and lead to poor quality of testing.
5	Not recording non-reproducible defects	Many times tester came across bugs during random / exploratory testing which appeared on specific configurations and are non-reproducible. This made testing task extremely tedious and time consuming, as many times there would be random hangs in product.
6	Tedious manual verification and testing the complete application	Even though this led developers on displaying specific interpretation of results, this has to be done on wide range of datasets and is a repetitive work. Also to test each and every combination was challenging.
7	Interdependencies of components in the software	Since the software was complex with different components, the changes in one part of software often caused breaks in other parts of the software. Pressure to handle the current functionality changes, previous working functionality checks and bug tracking.
8	Testing always under time constraints	Often there was a slippage in other phases of the project and thus reduced time for testing as there was a committed end date to customer. It was also observed that the tester could simply focus on task completion and not on the test coverage and quality of work. This testing activity was taken up as last activity in project life cycle and there was always a pressure to squeeze testing in a short time.
9	<p>a. Test Systems inadequacy &amp; lack of dedicated resources for test team.</p> <p>b. Under estimating testing efforts in project efforts</p>	<ul style="list-style-type: none"> <li>• Testing time was affected because of lack of dedicated test systems given to test team, the testers got assigned to test multiple modules and the developers were finally moved on the testing job.</li> <li>• Test engineers were forced to work at odd hours/weekends as the limited resources were in control of the development team and test engineers were given a lower priority during allocation of resources.</li> <li>• Testing team was not involved during scoping phase and the testing team's efforts were typically underestimated. This led to lower quality of testing as sufficient efforts could not be put in for the same.</li> </ul>

10	The involvement of test team in entire life cycle is lacking	Test engineers were involved late in the life cycle. This limited their contribution only to black box testing. The project team didn't use the services of the test team for the unit as well as integration testing phases. Due to the involvement testers in the testing phase, the test engineers took time to understand all the requirements of the product, and were overloaded and finally were forced to work many late hours.
11	Problems faced to cope with attrition	Few Key employees left the company at very short career intervals. Management faced hard problems to cope with attrition rate. New testers taken into project required project training from the beginning and as this is a complex project it became difficult to understand thus causing delay in shipping date.
12	Hard or subtle bug remained unnoticed	Since there was a lack of skilled testers and domain expertise, some testers concentrated more on finding easy bugs that did not require deep understanding.
13	Lack of relationship with the developers & no documentation accompanying releases provided to test team	It was a big challenge. There was no proper documentation accompanying releases provided to the test team. The test engineer was not aware of the known issues, main Features to be tested, etc. Hence a lot of effort was wasted.
14	Problems faced to cope up with scope creep and changes to the functionality.	Delays in implementation date because of lot of rework. Since there were dependencies among parts of the project and the frequent changes to be incorporated, resulted many bugs in the software.

#### V. ANALYSES OF THE ABOVE PROBLEMS THAT MADE THE PROJECT DEEMED TO BE QUITE CHALLENGING

One of the things that we've noticed over the past few years of working on software delivery projects is that the most challenging projects face the problems related to the defects in the product which is about to be delivered to the customer. Project success or failure depends largely on how you address the above basic issues. When analysed the above challenges in table 1 carefully it can be seen that a major source of schedule slippage and cost overruns is the fact that the applications have so many bugs that they won't work or cannot be released. So this stands as the primary reason for software delays and cancellations because of more bugs or errors whose elimination can absorb more than half of the effort on really large software applications. Another important reason for the software failure is the increase in unplanned creeping requirements (scope creep) added during design and coding phases tend to generate far more than their share of errors. These slow down testing and stretch out schedules. So, eventually we decide that we've had enough of feeling sorry for ourselves and finally move to the stage of acceptance where we can start working on solutions to the problems that we can solve instead of focusing on the things that are out of our control. Even CMMI level5 organisations are still trying to implement good engineering practices in both Verification and the Validation processes. Also these organizations pose a major challenge in delivering the projects/products within committed timelines and meeting customer expectations. In making a project success the test team must be competent enough in all the 3 areas Domain Knowledge, Testing skill, and Technical expertise.

Going through the series of lessons learnt from some of the recent major computer system failures caused by software bugs [2][3][4] and the analysis of the above problems in table 1 made us to make things better for ourselves to apply in the projects ahead instead of focusing on all the things that are out of our control. This activity helped us a lot by ensuring maximum learning from the situation so that we would be able to handle something similar more effectively in the future.

The following subsections provide summarizes actionable lessons learned from a comprehensive survey examining testing practices.

##### A. Initiative to implement Risk Based Validation Approach

Risk based validation approach was adapted to conduct a thorough risk analysis to identify any possible defects associated with a particular phase of project life cycle and rigorously tested the areas of the software system that pose the greatest risk to product quality. By conducting this activity the overall validation costs were reduced and efficiency was increased.

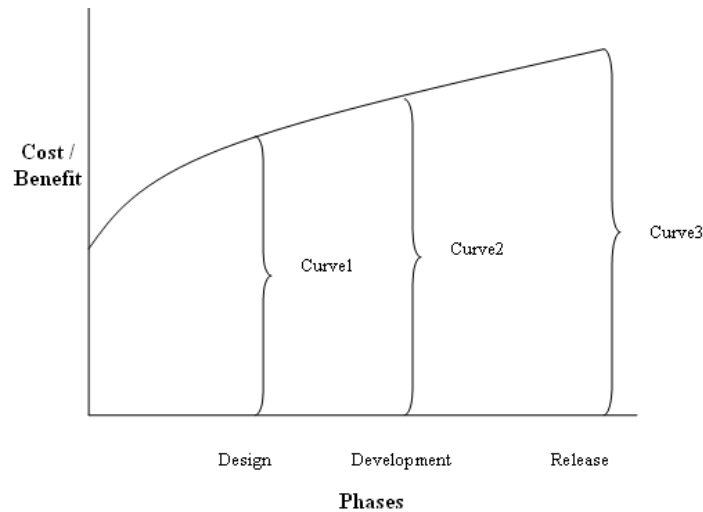


Figure 1. Cost/Benefit and Risk assessment

Figure 1 above shows cost/benefits due to risk assessments during various phases of development life cycle. In the above diagram,

Curve1: Identify and remediate an area of risk prior to design (requirements) is least expensive

Curve2: Identify and remediate an area of risk during design and development is more expensive

Curve3: Remediate an area of risk after release is most expensive

In any project, identifying, evaluating, controlling, and mitigating risk may allow for shorter development cycle. If the delivery time is too short for a thorough testing of all functions, the focus had to be put on those areas representing the largest risk if a fault occurred. The project needs a methodology to identify the most critical areas to test and utilization of limited resources during test preparation and test execution. In the Traditional Approach to testing, the test strategy document identifies system test plan, test specification, test cases, and test scripts and Implementing a structure like the one above is very time consuming. Because of this either the project would be delayed, or the test planning process had to be changed. The risk based test approach is highly dependent on using qualified testers, i.e. testers with experience within the application area and preferable with experience within the test environment [5]. The reason is that the tester himself will build the actual test scripts during test execution, including test data. Another criteria for success for the risk based approach was an efficient, dynamic and flexible test organization where it is essential that the testers do prioritize with regard to which faults to correct first and the functions need to be tested most based on the risk analysis and not on which fault is most convenient for the developers to fix.

Risk strategy → Risk Identification and Risk Assessment → Risk Mitigation and Risk Reporting

#### Step 1 - Risk Strategy

- i. Functionality Testing: The project will test all functionality in the application.
- ii. All test cases (“what to test”) will be documented prior to test start and will be available for the customer to review.
- iii. Only highly qualified testers, i.e. Subject Matter Expert or Domain Expert experienced in the application area, were to be utilised for testing, and the testers will be responsible for planning all “test shots”, including providing test data and documenting the executed tests. (Tools can be used for documenting the tests).

#### Step 2 - Risk Identification and Risk Assessment

The project (Project Manager, SME and experienced testers) will do a risk analysis together with the customer to identify those areas of highest risk, either to the customer or to the project.

#### Step 3 - Risk Mitigation and Risk Reporting

- i. Based on the Risk Analysis, the project should focus on “extra testing” in those areas of highest risk.
- ii. “Extra testing” will be planned and performed by a specialist in the application area that is not involved in the functionality testing.

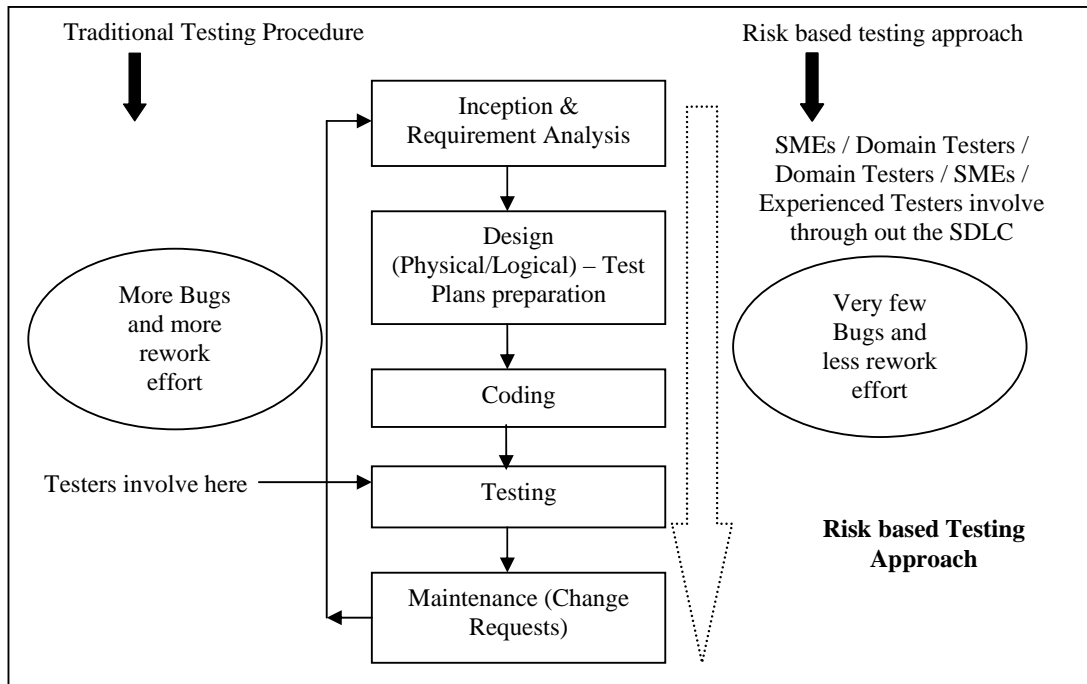


Figure 2. Comparison of Traditional and Risk Based Testing Practices

**B. Consideration of Test team to be involved in entire SDLC [6]**

The Defects generated out of requirements get carried on to next phase due to the insufficient review process. Due to late involvement of testers and frequent changes in requirements the test cases designed may not be of adequate coverage and force testers to determine expected results which may be a challenging issue. Testing has to be well planned, prepared, designed, executed and tracked taking into consideration on the risks and timelines. A complex software development requires including testing as early in the project as possible (from project’s inception itself). In the old Waterfall lifecycle world, testing was typically included in the coding phase right before they were needed. Even today the Requirements Phase of a project generates most of the defects, say over 50% of the defects are injected during the requirements phase. Requirements Phase is important since this phase forms the foundation for the product that more of the later work will build on. The amount of rework to fix assumptions or undiscovered issues can have a major impact on a project [7]. The above issues convey the need for testers to be active participants from the requirements phase itself till the deployment phase and should not be passive recipients of requirements in the form of a Use Case or a requirements document. Testing has always been challenging in the software development life cycle of a product irrespective of any area of domain. Requirements that are important to the business are considered as high priority and requirements that make development uneasy are considered to be of high risk. Most of the bugs in software are due to incomplete or inaccurate functional requirements. If the requirements are not correct there may be fundamental issues that are not surfaced until the later phases, which may require complete redesign of major portions of the product.

**C. Involvement of testers early in the project**

Usually testers understanding on the requirements will be limited and test cases designed may not be robust. Participation from Testing in requirements elicitation sessions may result in fewer changing requirements throughout the project, and fewer design problems [7]. Considering the fact that the early involvement during the project initial phase provides ample time for testers to gain good understanding on the scope and also contribute in refining good requirements, the testers were involved along with all other stakeholders right from the beginning of the lifecycle during the development of electronic medical records module. The test team was well aware of the client requirements and the testing

process was intended towards the actual requirements of the client. This also enabled the testers understanding the requirements and thus could jump-start on Test Case creation and could create several test cases. Understanding of the requirements and the interdependencies between requirements made it easier for testers to set test case to business requirement traceability.

Requirements review was a distinct activity done by the Domain Experts (DEs) / Subject Matter Experts (SMEs) and this activity ensured that testing of requirements was performed and it was performed correctly. This activity really helped occurrence of the major issues during later phases as the issues were caught during early phases of development.

*D. Importance to be given to domain testers in the review process*

Looking at the current scenario from the industry it is seen that the testers are expected to have both technical testing skills as well either need to be from the domain background or have gathered domain knowledge. Experienced professionals certainly have the advantage of domain and testing experience, have better understanding of different issues and can deliver the application better and faster. These domain generalists / domain experts with domain knowledge attending requirements review sessions will result in clearer and concise requirements [8]. It is observed from our experience that the following requires distinct edge of domain knowledge:

- i. Mobile application testing.
- ii. Healthcare Testing
- iii. Wireless application testing
- iv. Banking applications
- v. Network testing

It is difficult to test such applications without knowledge of specific domain. So, in our Electronic Medical Records (EMR) project the test engineers with testing knowledge and good understanding of clinical usage as used by the end users (Physicians, Nurses, and Lab technicians) at hospitals were deployed for testing along with the developers. These Experienced domain test engineers contributed more towards implementing the quality of the product and could execute more test cases effectively simulating the end user actions which were distinctly a big advantage. The project had implemented a good practice by involving Domain Experts during requirements phase they could use their own methods to uncover the unspecified requirements and also at the time of preparation of Software Requirements Specification document (SRS), so that they had their own understanding of the requirements that were specified. So this practice makes sure to avoid the irrelevant requirements in first phase of the project development cycle.

*E. Involvement of experienced reviewers in project life cycle*

The test team must be very experienced (or at least one lead tester per project) to assure high quality test products/results. The key advantage of an IV&V approach is that the IV&V team may be easier to position as a peer to development team. For this the IV&V group comprises a pool of experienced resources that are shared across applications and projects. Since the group does not belong to any particular application or project, they are less likely to report to development or project managers and insulated from the project management visibility. Gaining a voice in higher-level management decisions has value in many dimensions. This structure also helps to maintain a central repository for test plan, test cases and other documents. At present it is felt that test engineers must be trained and well equipped with standards like DICOM, HIPAA, IHE, HL7 etc. for testing the product for interoperability, connectivity and security aspects including ISO/CMMI processes and should realize that the ISO/CMMI processes have given the engineering rigor needed to achieve quality not merely the process steps.

## VI. CONCLUSION

Domain expertise must remain within the project throughout its lifecycle [9]. The Management usually feels that once up-front analysis is complete, the analysts have less work, and their role becomes more passive. This cannot be justified as these people are valuable to the business and their effort is considered as various activities of the project life cycle. A solution that often works well is to keep a few domain experts assigned full time, and give them the permanent role of facilitator. In this capacity, they perform domain analysis, and execute all change requests to requirements specifications. They also develop user-oriented test plans, and construct system documentation. Their role therefore remains an

active one, and their knowledge about the application, and contacts within the organization, can still be tapped when questions arise during development. When a project is low on funding or over budget, the first important task that is cut is 'Testing'. But limiting testing of a solution will increase the chance of system failure or unknown bugs that can cripple your business.

Future research should address a set methodology for doing product/project inspections, as until now, there has been no set methodology for doing reviews/inspections, no requirement that reviews be conducted under defined circumstances or at specified points, and no precise qualifications required for reviewers.

#### REFERENCES

- [1] David F. Rico, V&V Lifecycle Methodologies.
- [2] "Major computer system failures caused by software bugs", <http://forums.sureshkumar.net/latest-tech-news- innovations/11293-some-recent-major-computer-system-failures-caused-software-bugs.html>.
- [3] "Major computer system failures caused by software bugs", <http://www.zdnet.com/blog/projectfailures/40-it-failures-caused-by-software-bugs/427>.
- [4] "Recent major computer system failures caused by software bug", <http://geekswithblogs.net/ejohnson/archive/2004/05/28/5514.aspx>
- [5] Hans Schaefer, Risk based testing [Online], Available: <http://home.c2i.net/schaefer/testing.html>, 1999.
- [6] "Involve Testing Throughout the SDLC" [Online], Available: [www.silverpath.com](http://www.silverpath.com).
- [7] Gaurav Bhatia, "Why tester's involvement is important in requirement gathering?", <http://blogs.ebusinessware.com/2009/05/22/why-tester%E2%80%99s-involvement-is-important-in-requirement-gathering>
- [8] Michael J. Johnson and E. Michael Maximilien, IBM and Chih-Wei Ho and Laurie Williams, North Carolina State University, "Incorporating Performance Testing in Test-Driven Development", May/June 2007 IEEE
- [9] Marvin V. Zelkowitz, Ioana Rus, "The Role of Independent Verification and Validation in Maintaining a Safety Critical Evolutionary Software in a Complex Environment: The NASA Space Shuttle Program", NASA Subcontract No. 93-393B-FUSA from the NASA/IVV facility in Fairmont, WV to the Fraunhofer Center, Maryland.

#### AUTHORS PROFILE



Dr. K. Nageswara Rao is Currently working as Professor & Head in the Department of Computer Science Engineering, Prasad V. Potluri Siddhartha Institute of Technology, Kanuru, Vijayawada-7. He has an excellent academic and research experience. He has contributed various research papers in the journals, conferences of International/national repute. His area of interest includes Artificial Intelligence, Software Engineering, Robotics, & Datamining.



Mr. A. Pathanjali Sastri is currently pursuing Ph.D Computer Science from Raylaseema University, Kurnool. He is working as a Lecturer in Velagapudi Siddhartha Engineering college since 2008 and has 10 years of Industrial experience prior to this. He has published papers in reputed journals/International conferences recently. His area of interest includes Software Engineering, Quality Assurance, Artificial Intelligence and RDBMS.