JCSE International Journal of Computer Sciences and Engineering Open Access

Research Paper Volume-3, Issue-1 E-ISSN: 2347-2693

# Analysis using Non-Functional Static Testing Framework

Priyanka R. Bhuyar[1*] and .A.D.Gawande[2]

[1]M.E. IInd Year(CSE),Sipna College of Engineering & Technology, Amravati
[2]HOD (CMPS), Department of computer science & Engineering, Sipna College of Engineering & Technology, Amravati

www.ijcaonline.org

Received: Dec /26/2014 Revised: Jan/8/2015 Accepted: Jan/20/2015 Published: Jan/31/2015

***Abstract-*** Static Testing Techniques are widely used throughout SDLC and proved very effective increasing effectiveness of testing by uncovering and preventing defects at early stage. Static Testing techniques for functional aspects of software are well defined and matured compared to static testing techniques for Non functional aspects of software.

In this project, we designed Non Functional Static Testing Framework with details such as processes, roles, test scenarios, implementation guidelines. Framework helps to use correct NF static technique at correct place (which technique to use, where to use it in SDLC, how to use). We strongly believe that NF static testing framework is a novel idea and fulfill several goals in increasingly complex non-functional domain, thereby reducing business risks at optimal cost and schedule.

Key words:- SDLC, Static Test Cases, 3 tier Architecture

## I. Introduction

Software testing is performed to evaluate and guarantee the level of quality of software as per the specified requirements. In testing, a program or system is executed with required input data to find the errors [1][2]. Along with the validating functionality of applications, software testing is also used to test other non-functional quality factors like performance, reliability, usability, integrity, security, capability, efficiency, portability, maintainability, compatibility, availability etc [3].

At present, static testing is formally and widely used as part of functional testing and proved to be very useful to increase effectiveness of testing [4]. Static testing for Non-Functional aspects of application are not being widely used compared to functional aspects of applications. Here in this project, we studied non functional static techniques being used at present and develop the non functional static testing framework which can be used in industry. Framework covers different test scenarios to be applied at during SDLC phases along with process, roles and pre-requisites involved.

Non-functional requirements are defined from end user perspective considering all the components involved in the system. However, static testing needs to be carried out at component level along with E2E level. The expectations from most of the customers and organizations have now changed significantly from a non-functional perspective.Major scenarios which have added complexities and associated challenges to non functional aspects are as below:

- Ensuring consistent performance metrics for very large number of users across multiple geographies which use heterogeneous channels including smart devices, laptops and the cloud
- Applications getting increasingly complex with large number of heterogonous integration points/data sources
- New latest technology applications such as Mobile applications, Cloud applications, Analytics, Big Data Solutions, enterprise applications, Internet of things applications and Line of Business specific applications (Telecom, Finance, Online Retail etc.) are getting added in the IT landscape.
- Focus in Non-Functional Quality Assurance is changing from testing to business assurance.
- In addition to these, demand for optimizing cost, scope and schedule is high in an aggressive market scenario.

Non-functional static testing framework will help to find some of the major non-functional issues early in software development life-cycle.The work in this project involves study of existing informal methods and design Core Non-functional Static Testing Framework for performance, scalability, reliability and availability. The core framework include –

- The NF static testing techniques to be used in different phases of SDLC i.e. Requirement, Plan, Design, Development, Testing, Deployment and Production.
- Test Cases, Metrics, Checks for NF static Testing techniques of framework
- The different roles involved and activities by them at each phase of SDLC for different NF static testing techniques.
- The guidelines to select and use particular NF static testing technique based on its merits, demerits.

The framework helps to face increasing complexities and challenges in non-functional space along with optimized cost, schedule and quality.

The framework also helps to select Testing teams Performance metrics based on technology and servers during test planning phase.

## II. Software Application 3 tier Architecture

Performance bottleneck identification is a major task in a 3-tier web applications performance testing because there can be various profiled reasons behind a slow performing web application.

In software engineering, different client-server architectures are used for web application development. These architectures logically differentiate among data presentation, application processing and data management functions. 3-tier architecture (web server, application server and database server) is the most famous N–tier client-server architecture used for web applications at an enterprise level.

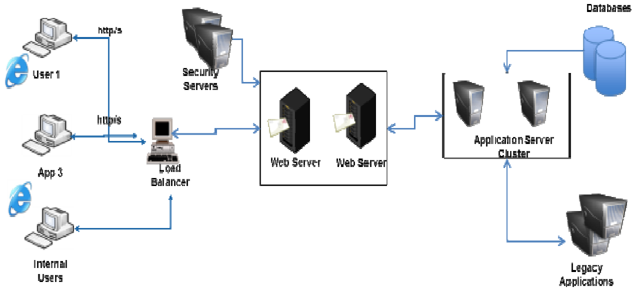

*Fig-2.1. Typical 3-Tier Web Application Architecture*

Web server is 1st tier in a 3-tier architecture generally consists of low capacity computer(s) that receive the user requests and send them to required downstream server and send received results to users. Application server is 2nd tier in a 3-tier architecture having one or more medium to large capacity computer(s) that receives user requests from the web server, apply business logic on them and send them back to the web server. Database server is in the last tier of a 3-tier client-server architecture which normally consists of a high capacity computer with stand-by facility that manages database access to facilitate user data requests.

### 2.1 Performance Aspects of a 3-tier Architecture

Performance aspects and potential performance bottlenecks which can occur at above discussed architectural components namely web server, application server, database Server and network.

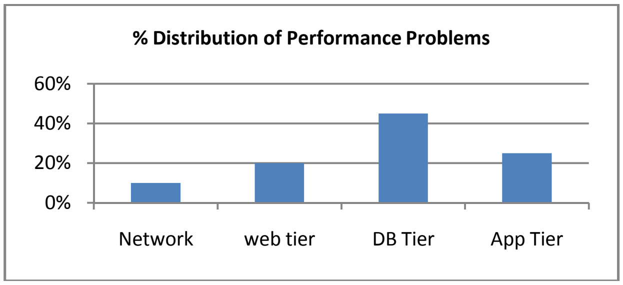

*Fig2.2: Performance Problems Distribution among the components of 3 - tier architecture*

The above chart shows the distribution of performance problems among different tiers of architecture. Hence we need to focus on performance aspects different tiers of architecture while using Non-functional static testing techniques.

## III. Static Test Cases

In this section, Non-Functional static testing cases which can be applied at each phase of SDLC are described. These test cases are executed by non-functional testing team in those phases, pass/fail criterion is applied and defects are raised/assigned to responsible teams for that phase.

### 3.1 Performance Static Test Cases

#### 3.1.1 Performance Static Test Cases for Requirement Phase of SDLC

| Test No | Test Case Description | Measures i.e. Pass/Fail Criteria |
|---|---|---|
| 1 | Are there any implicit or explicit performance requirements? | if yes, execute next test cases. |
| 2 | Are performance requirement documented in requirements documented? | if no, log a defect and assign to requirements teams. |
| 3 | Does the requirement quantify what performance improvement is required in this release or state existing performance should be sustained? | if yes, pass the test, if no, log a defect and assign to requirements teams. |
| 4 | Is the requirement of User Response Times quantified adequately? | Does each of these requirements clearly translate to one or more test cases? If yes, then this test should pass else fail. |
| 5 | Is the requirement of Throughput quantified adequately? | Does each of these requirements clearly translate to one or more test cases? If yes, then this test |

| | | |
|---|---|---|
| | | should pass else fail. |
| **6** | Is the requirement of Concurrent users, Max Users, Avg Users quantified adequately? | Do each of these requirements clearly translate to one or more test cases? If yes, then this test should pass else fail. |
| **7** | Is the requirement of Peak Window and Off-Peak window quantified adequately? | Do each of these requirements clearly translate to one or more test cases? If yes , then this test should pass else fail. |
| **8** | Is the requirement of Batch job window and Batch Size quantified adequately? | Do each of these requirements clearly translate to one or more test cases? If yes , then this test should pass else fail. |
| **9** | Is the requirement of Peak Number of Total system Transactions and Off-Peak Number of Total system Transactions quantified adequately? | Do each of these requirements clearly translate to one or more test cases? If yes , then this test should pass else fail. |
| **10** | Is the requirement of Peak Number of Transactions and Off-peak Number of Transactions quantified adequately for each business critical use case or scenario? | Do each of these requirements clearly translate to one or more test cases? If yes, then this test should pass else fail. |
| **11** | Are CPU Utilization limits defined for hardware components across the layers of architecture? | if yes, pass the test, if no, log a defect and assign to requirements teams. |
| **12** | Are Memory Utilization limits defined for hardware components across the layers of architecture? | if yes, pass the test, if no, log a defect and assign to requirements teams. |
| **13** | Are Disk I/O Utilization limits defined for hardware components across the layers of architecture? | if yes, pass the test, if no, log a defect and assign to requirements teams. |
| **14** | Is backend load if any along with main call flows considered? | if yes, pass the test, if no, log a defect and assign to components/ requirements teams. |

### 3.1.2 Performance Static Test Cases for Design Phase of SDL

| Test No | Test Case Description | Measures i.e. Pass/Fail Criteria |
|---|---|---|
| **1** | Does the design cover all implicit and explicit performance requirements? | If yes, pass the test else fail. |
| **2** | Is the design information satisfactory enough to validate by testing? | If yes, then this test should pass else fail; check for evidence. |
| **3** | Does the design(s) describe in sufficient detail and trace back to one or more of the performance requirements? | If yes, then this test should pass else fail; check for evidence. |
| **4** | Does the solution support the number of concurrent users, max users, avg users and user accounts specified? | If yes, then this test should pass else fail; check for evidence. |
| **5** | If the solution uses existing platforms / technology, does the solution meet stated performance targets? | If yes, then this test should pass else fail; check for evidence. |
| **6** | Does the solution meet the total transactional volume demand and at the same time deliver the quality of service? | If yes, then this test should pass else fail; check for evidence. |
| **7** | Is the solution compliant with organization's stated requirements for the mix of business transactions that can be handled simultaneously / concurrently? | If yes, then this test should pass else fail; check for evidence. |
| **8** | Is the solution compliant with organization's stated requirements for the mix of Users that can be handled simultaneously / concurrently? | If yes, then this test should pass else fail; check for evidence. |
| **9** | If the solution uses existing platforms / technology, does the solution state the change, if any, to the volume of concurrent transactions that can be handled? | If yes, then this test should pass else fail; check for evidence. |
| **10** | Does the solution meet agreed response times does the response time of the solution to Users meet agreed standards? | If yes, then this test should pass else fail; check for evidence. |
| **11** | Does solution specifies batch sizes, frequency, time of day for batch jobs processing? | If yes, then this test should pass else fail; check for evidence. |

### 3.1.3 Performance Static Test Cases for Testing Phase of SDL

| Test No | Test Case Description | Measures i.e. Pass/Fail Criteria |
|---|---|---|
| 1 | Are all requirements risk assessed and signed off by stakeholders including (a) the developers and (b) the Business Users? | If yes, then this test should pass else fail; check for evidence. |
| 2 | Is the requirements risk assessment matrix published? Is the agreed criteria for "inclusion of requirements for testing" clearly written? | If yes, then this test should pass else fail; check for evidence. |
| 3 | Are the scenarios/behaviors that will be tested / not tested or partly tested clearly listed? | If yes, then this test should pass else fail; check for evidence. |
| 4 | Does risk assessment include all new and regression tests in scope? | If yes, then this test should pass else fail; check for evidence. |
| 5 | Is scalability of Test environment compared to production environment considered during test design? | If yes, then this test should pass else fail; check for evidence. |
| 6 | Have you referred production performance (volumes, response time, throughput, max users etc.) issues while test planning? | If yes, then this test should pass else fail; check for evidence. |
| 7 | Are baseline results till date available (mostly the regression or last release test results)? | If yes, then this test should pass else fail; check for evidence. |

## 3.2 Availability Static Test Cases

### 3.2.1 Availability Static Test Cases for Requirement Phase of SDLC

| Test No | Test Case Description | Measures i.e. Pass/Fail Criteria |
|---|---|---|
| 1 | Are there any implicit or explicit Availability requirements? | if yes, execute next test cases. |
| 2 | Are Availability requirement documented in requirements documented? | if no, log a defect and assign to requirements teams. |
| 3 | Does the requirement quantify what Availability improvement is required in this release or state existing performance should be sustained? | if yes, pass the test, if no, log a defect and assign to requirements teams. |
| 4 | Is the requirement of Service Availability is defined e.g. 99.9% availability? | Do each of these requirements clearly translate to one or more test cases? If yes , then this test should pass else fail. |
| 5 | Are number of refusals in peak hour defined as a part of availability requirements? | Do each of these requirements clearly translate to one or more test cases? If yes, then this test should pass else fail. |
| 6 | Are the service hours for the public Internet service e.g.<br>all informational + transactional pages must be available 24 hours 7 days a week? | Do each of these requirements clearly translate to one or more test cases? If yes , then this test should pass else fail. |
| 7 | Is the requirement of continuous data replication between the two data centers to avoid permanent data loss defined? | Do each of these requirements clearly translate to one or more test cases? If yes, then this test should pass else fail. |
| 8 | Is the requirement of throttling throughput in situations of excessive load defined? | Do each of these requirements clearly translate to one or more test cases? If yes , then this test should pass else fail. |
| 9 | Are the priorities to transactions assigned in the event of a reduced transactional service capability e.g. immediate payments and uploading batched payments should take priority over other transactions. | Do each of these requirements clearly translate to one or more test cases? If yes, then this test should pass else fail. |
| 10 | Is the requirement of site fail-over from the primary site to the secondary site defined? | Do each of these requirements clearly translate to one or more test cases? If yes , then this test should pass else fail. |
| 11 | Does the requirement include Recovery Point Objectives and Recovery Time Objectives? | Do each of these requirements clearly translate to one or more test cases? If yes, then this test should pass else fail. |

| | | |
|---|---|---|
| **12** | Do the requirements include Disaster Recovery Point Objectives and Disaster Recovery Time Objectives which are to be agreed? | Do each of these requirements clearly translate to one or more test cases? If yes , then this test should pass else fail. |

### 3.2.2 Availability Static Test Cases for Design Phase of SDLC

| Test No | Test Case Description | Measures i.e. Pass/Fail Criteria |
|---|---|---|
| **1** | Does the design cover all implicit and explicit Availability requirements? | If yes, execute next test cases. |
| **2** | Is the design information satisfactory enough to validate by testing? | If yes, then this test should pass else fail; check for evidence. |
| **3** | Does the design(s) describe in sufficient detail and trace back to one or more of the Availability requirements. | If yes, then this test should pass else fail; check for evidence. |
| **4** | If the solution uses existing platforms / technology, does the solution meet stated Availability targets? | If yes, then this test should pass else fail; check for evidence. |
| **5** | Does the solution ensures to achieve a minimum of stated service hour availability (e.g. 99.9%). | If yes, then this test should pass else fail; check for evidence. |
| **6** | Does the solution support to achieve number of refusals in peak hour defined as a part of availability requirements? | If yes, then this test should pass else fail; check for evidence. |
| **7** | Does the solution support to achieve defined service hours for the public Internet service e.g. all informational + transactonsal pages must be available 24 hours 7 days a week? | If yes, then this test should pass else fail; check for evidence. |
| **8** | Does the solution will continue to support peak capacity demand on failure of a single component at the primary site? | If yes, then this test should pass else fail; check for evidence. |
| **9** | Does the solution includes impact in case of exceeding threshold of concurrent Users (e.g. runs slow or denies access)? | If yes, then this test should pass else fail; check for evidence. |
| **10** | Does the solution clearly state any action triggered, if the number of users is about to breach the max threshold? | If yes, then this test should pass else fail; check for evidence. |

### 3.2.3 Availability Static Test Cases for Testing Phase of SDLC

| Test No | Test Case Description | Measures i.e. Pass/Fail Criteria |
|---|---|---|
| **1** | Are all requirements risk assessed and signed off by stakeholders including (a) the developers and (b) the Business Users? | If yes, then this test should pass else fail; check for evidence. |
| **2** | Is the requirement risk assessment matrix published? Is the agreed criteria for "inclusion of requirements for testing" clearly written? | If yes, then this test should pass else fail; check for evidence. |
| **3** | Are the scenarios/behaviors that will be tested / not tested or partly tested clearly listed? | If yes, then this test should pass else fail; check for evidence. |
| **4** | Does risk assessment include all new and regression tests in scope? | If yes, then this test should pass else fail; check for evidence. |

## 3.3 Reliability Static Test Cases

### 3.3.1 Reliability Static Test Cases for Requirement Phase of SDLC

| Test No | Test Case Description | Measures i.e. Pass/Fail Criteria |
|---|---|---|
| **1** | Is there any explicit or implicit Reliability and Contingency requirement? | if yes, execute next test cases. |
| **2** | Are Reliability and Contingency requirement documented in requirements documented? | if no, log a defect and assign to requirements teams. |
| **3** | Does the requirement state that Failover should be implemented as a part of reliability and contingency measures? | Is the expected outcome of this requirement clear? Does each of these requirements clearly translate to one or more test cases? If yes, then this test should pass. Else fail. |

| | | |
|---|---|---|
| 4 | Does the requirement state that Load Balancing should be implemented as a part of reliability and contingency measures? | Is the expected outcome of this requirement clear? Do each of these requirements clearly translate to one or more test cases? If yes, then this test should pass. Else fail. |
| 5 | Does the requirement state continuous System Uptime? | Is the expected outcome of this requirement clear? Do each of these requirements clearly translate to one or more test cases? If yes, then this test should pass. Else fail. |
| 6 | Does the requirement state fallback option during downtime service? | Is the expected outcome of this requirement clear? Do each of these requirements clearly translate to one or more test cases? If yes, then this test should pass. Else fail. |

### 3.3.2 Reliability Static Test Cases for Design Phase of SDLC

| Test No | Test Case Description | Measures i.e. Pass/Fail Criteria |
|---|---|---|
| 1 | Does the design(s) describe in sufficient detail and trace back to one or more of the requirements? | If yes, then this test should pass else fail; check for evidence. |
| 2 | Does the design cover all implicit and explicit requirements? | If yes, then this test should pass else fail; check for evidence. |
| 3 | Is the design information satisfactory enough to validate by testing? | If yes, then this test should pass else fail; check for evidence. |
| 4 | Does the solution meet Failover requirements as stated? | If yes, then this test should pass else fail; check for evidence. |
| 5 | Does the solution meet Load Balancing requirements as stated? | If yes, then this test should pass else fail; check for evidence. |
| 6 | Does the solution meet Continuous system uptime requirements as stated? | If yes, then this test should pass else fail; check for evidence. |
| 7 | Does the solution meet to deliver failback options as stated? | If yes, then this test should pass else fail; check for evidence. |

### 3.3.3 Reliability Static Test Cases for Testing Phase of SDLC

| Test No | Test Case Description | Measures i.e. Pass/Fail Criteria |
|---|---|---|
| 1 | Are all requirements risk assessed and signed off by stakeholders including (a) the developers and (b) the Business Users? | If yes, then this test should pass else fail; check for evidence. |
| 2 | Is the risk assessment matrix published? Is the agreed criteria for "inclusion of requirements for testing" clearly written? | If yes, then this test should pass else fail; check for evidence. |
| 3 | Are the scenarios/behaviors that will be tested / not tested or partly tested clearly listed? | If yes, then this test should pass else fail; check for evidence. |
| 4 | Is non availability/partial availability of some reliability and contingency mechanism in Test environment compared to production environment considered during test design? | If yes, then this test should pass else fail; check for evidence. |
| 5 | Have you referred Production reliability and contingency issues considered while test planning? | If yes, then this test should pass else fail; check for evidence. |

## 3.4 Scalability Static Test Cases

### 3.4.1 Scalability Static Test Cases for Requirement Phase of SDLC

| Test No | Test Case Description | Measures i.e. Pass/Fail Criteria |
|---|---|---|
| 1 | Are there any implicit or explicit Scalability requirements? | if yes, execute next test cases. |
| 2 | Is Scalability requirement documented in requirements documented? | if no, log a defect and assign to requirements teams. |
| 3 | Does the requirement quantify what Scalability improvement is required in this release or state existing performance should be sustained? | if yes, pass the test, if no, log a defect and assign to requirements teams. |
| 4 | Does the requirement quantify growth of number of concurrent | if yes, pass the test, if no, log a defect and assign to |

| | | |
|---|---|---|
| | users, avg users over next 1-3 years for applications? | requirements teams. |
| **5** | Does the requirement quantify growth of products, accounts, items etc over next 1-3 years for applications? | if yes, pass the test, if no, log a defect and assign to requirements teams. |
| **6** | Does the requirement specify growth in number of branches, cities, countries and geographies for next 1-3 for a business? | if yes, pass the test, if no, log a defect and assign to requirements teams. |
| **7** | Does the requirement specify new channels of business (mobile, social media, etc). | if yes, pass the test, if no, log a defect and assign to requirements teams. |

### 3.4.2 Scalability Static Test Cases for Design Phase of SDLC

| Test No | Test Case Description | Measures i.e. Pass/Fail Criteria |
|---|---|---|
| **1** | Does the design cover all implicit and explicit Scalability requirements? | If yes, pass else fail; check for evidence. |
| **2** | Is the design information satisfactory enough to validate by testing? | If yes, then this test should pass else fail; check for evidence. |
| **3** | Does the design(s) describe in sufficient detail and trace back to one or more of the performance requirements? | If yes, then this test should pass else fail; check for evidence. |
| **4** | Does the solution ensures that file, table and database sizes take into account room for anticipated growth? | If yes, then this test should pass else fail; check for evidence. |
| **5** | Does the solution have the capability to scale vertically and horizontally to cater to stated scalability requirements? | If yes, then this test should pass else fail; check for evidence. |
| **6** | Does the design describe solution in sufficient details to support growth of number of concurrent users, avg users over next 1-3 years for applications as stated? | If yes, then this test should pass else fail; check for evidence. |
| **7** | Does the design describe solution in sufficient details to support growth of products, accounts, items etc over next 1-3 years for applications? | If yes, then this test should pass else fail; check for evidence. |
| **8** | Does the design describe solution in sufficient details to support growth in number of branches, cities, countries and geographies for next 1-3 for a business? | If yes, then this test should pass else fail; check for evidence. |
| **9** | Does the requirement specify new channels of business (mobile, social media, etc?) | If yes, then this test should pass else fail; check for evidence. |

### 3.4.3 Scalability Static Test Cases for Testing Phase of SDLC

| Test No | Test Case Description | Measures i.e. Pass/Fail Criteria |
|---|---|---|
| **1** | Are all requirements risk assessed and signed off by stakeholders including (a) the developers and (b) the Business Users? | If yes, then this test should pass else fail; check for evidence. |
| **2** | Is the risk assessment matrix published? is the agreed criteria for "inclusion of requirements for testing" clearly written? | If yes, then this test should pass else fail; check for evidence. |
| **3** | Are the scenarios/behaviors that will be tested / not tested or partly tested clearly listed? | If yes, then this test should pass else fail; check for evidence. |
| **4** | Is scalability of Test environment compared to production environment considered during test design? | If yes, then this test should pass else fail; check for evidence. |
| **5** | Have you referred Production Scalability (volumes, response time, Throughput, concurrent users etc. ) issues while test planning? | If yes, then this test should pass else fail; check for evidence. |
| **6** | Are baseline results till date available (mostly the regression or last release test results)? | If yes, then this test should pass else fail; check for evidence. |

## IV. Conclusion

In this project work, study of common performance bottlenecks and factors contributing to those bottlenecks for different components of 3-tier application architecture is carried out. Non-functional aspects performance, reliability, availability and scalability are studied in detail and non-functional static testing framework is designed for these areas. Non-functional static test cases, pass/fail criterion are designed for Requirements, Design and

Testing phase of SDLC. This will provide one consolidated framework for Non- functional testing teams in IT companies. This helps to formalize process of non-functional testing in a project or a programme. In the absence of well-designed and documented framework, it was difficult for NF Testing teams to get involved in requirements phase and design phase for validation of non-functional aspects and raising defects against requirements and design teams.

## REFERENCES

[1] Quadri, S.M.K and Farooq,“3W’s of Static Software Testing Techniques”, Global Journal of Computer Science and Technology Volume 11, Issue 5, Version 1.0, April 2011.

[2] Quadri, S.M.K and Farooq, SU, “Software Testing – Goals, Principles, and Limitations”, International Journal of Computer Applications (0975 – 8887) Volume 6– No.9, September 2010.

[3] Vipin Saxena and Santosh Kumar, “Performance Computation Metrics for Object-Oriented Software Systems”, International Journal of Advanced Research in Computer Science, Volume 3, No. 6, Nov. 2012 (Special Issue)

[4] Kamna Gauri and Dipanwita Thakur, “Comparative Performance Evaluation Of Software Architectural Styles With UML”, International Journal of Advanced Research in Computer Science, Volume 3, No. 1, Jan-Feb 2012

[5] Shyam S. Pandeya and Anil K. Tripathi, “Techniques for Developing Testable Component Based Software: Similarities, Differences and Classification”, International Journal of Advanced Research in Computer Science, Volume 2, No. 2, May-June 2010.

[6] James H. Hill, Hamilton A. Turner, James R. Edmondson, and Douglas C. Schmidt, “Unit Testing Non-functional Concerns of Component-based Distributed Systems”, International Conference on Software Testing Verification and Validation April 2009.

[7] Myers, Glenford J., The art of software testing, Publication info: New York: Wiley, c1979. ISBN:0471043281

[8] Hetzel, William C. The Complete Guide to Software Testing, 2nd ed. Publication info: Wellesley, Mass.: QED Information Sciences, 1988. ISBN: 0894352423

[9] Roper, M., “Software Testing”, McGraw-Hill, 1994, 149 pages, ISBN 0-07-707466-1

[10] Somerville, I, “Software Engineering”, Pearson, 2008, 864 pages, ISBN 978-81-317-2461-3

[11] G. Denaro, A. Polini, and W. Emmerich. Early Performance Testing of Distributed Software Applications. ACM SIGSOFT Software Engineering Notes, 29(1):94–103, January 2004

[12] Review process, http://www.testresources.info/iseb-software-testing-reviewprocess.php

[13] IEEE, "IEEE Standard 610.12-1990, IEEE Standard Glossary of Software Engineering Terminology," 1990

[14] Prof. Dr. Cremers, Sascha Alda, “Organizational Requirements Engineering, Chapter 9, Non-functional Requirements”

[15] Dan Bergh Johnson, “ Non-functional requirements- how to get them in shape”, Colorado Software Summit, October 2006.