SindhUniv. Res. Jour. (Sci. Ser.) Vol. 47(2):313-320 (2015)

SINDH UNIVERSITY RESEARCH JOURNAL (SCIENCE SERIES)

Factors Affecting Software Quality in Legacy Software Life Cycle Models for Emerging Professionals

N. MAHOTO[++], A. SHAIKH, F. KHUHAWAR* A. AFTAB

Department of Software Engineering, Mehran UET Jamshoro, Sindh, Pakistan

Received 4th March 2015 and Revised 16th May 2015

Abstract: The next generation computer technology facilitates humans to work efficiently through computer machines. These machines need Quality software to perform required tasks appropriately. The Quality of software exhibits the characteristics of the software product, which can be achieved through various processes performed during software development lifecycle. When software is designed and built not following standard processes, eventually results in more chances of its failure. Sometimes, the chance of software failure increases because of having not well enough testing. To avoid failure of software, it is very much necessary to design and build the software using effective, standardized approaches and suitable software development models. This paper finds and identifies the key factors that may cause degradation of software quality and make software quality a challenging issue during the legacy software development lifecycle. The work is done with a survey conducted from development teams involved in the development of software in local cities of Pakistan. The survey results reveal the key factors and areas that affect software Quality in development lifecycle for emerging professionals in Pakistan.

Keywords: Software Engineering, Software Quality Assurance, Software Development, and Software Design.

1. INTRODUCTION

The next generation computer technology has changed life style; it has enhanced human efficiency exploiting computer machines, which require software as key element to accomplish predefined tasks. The computer machines make things better for humans if the software running on the machine performs all tasks smoothly. The success of software depends upon the requirements, which are supposed to be fulfilled. Thus the efficiency and success of the computer machine relies on software, which in return complies with user requirements or tasks.

Software Engineering discipline is concerned with designing, developing, testing and maintaining the software, and aims to produce Quality Software (Sommerville, 2000). The term Quality Software can be describe as software that is designed and built on time, within budget and satisfy the needs of its users. The development of Software involves many activities to produce Quality Software such as Analysis, Planning, and Design etc. **(Fig.1)** represents Software Development Lifecycle amongst various software process models.

- Waterfall Model (Royce, 1987)
- Boehm's Spiral Model (Boehm, 1988)
- Rational Unified Process (RUP) (Jacobson, Booch and Rumbaugh, 1998)
- RAD (Rapid Application Development) Model (Whitten, Bentley and Dittman, 2003)
- Agile (Beck *et al*., 2001)

The Quality of software exhibits the characteristics of the software that can be achieved through various processes performed in the development of software application. Although improvements are made to scale quality but when things go wrong, software fails to satisfy the needs (Pressman, 2001). Software, designed and built with non-standard processes, has more chances of failure. Sometimes, improper testing increases its failure chances. To avoid failures, designing and building of software using effective, standardized approaches and suitable software development models are needed.

In recent years much effort has been done to produce Quality Software to achieve better results in terms of productivity and efficiency of the organizations equipped with Information Technology (IT) solutions. Software, whilst works like backbone for such organizations. These organizations enjoy better productivity and scalable growth. In-fact Quality Software enhances efficiency of functional operations within organization, reduces time and cost, and increases customer trust. If the quality of software is poor or does not fully meet the intended requirements causes failure of entire effort performed in the development. The processes and activities performed in the development of software are interconnected to each other therefore defect of one activity may leave impact on other activities. Hence software must possess quality attributes such as: Reliability, Correctness, Integrity, Efficiency, and Maintainability. Moreover, many

[++]Corresponding Author Email: naeemmahoto@gmail.com; Tel.: +92-333-7538991

* Institute of Advanced Research Studies in Chemical Sciences, University of Sindh, Jamshoro, Sindh, Pakistan

organizations rely on Quality Software to increase productivity and scalability (Pressman, 2001), (Sommerville, 2000).

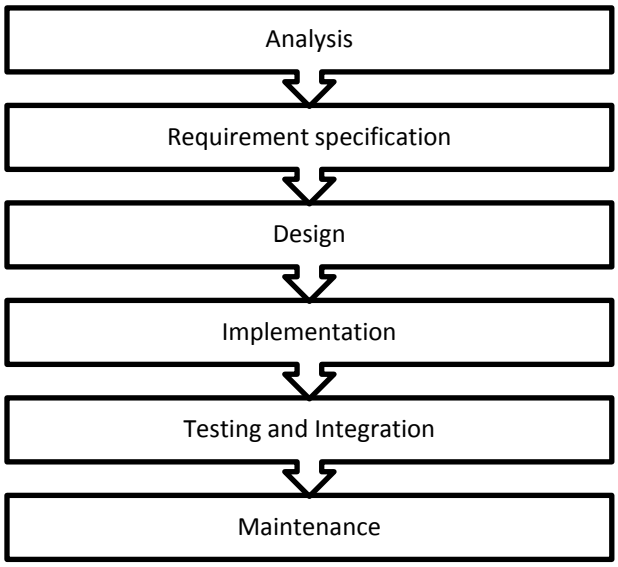

Fig.1 : Software Development Lifecycle

This paper addresses the factors that may degrade the software Quality during its development lifecycle. The research is carried out by conducting a survey from development teams of major cities of Pakistan to analyze the factors that impact software Quality. The survey results report the major areas and factors that cause the poor Quality of software while it is in development phase.

The rest of the paper is organized as Section 2 describes related work, Section 3 discusses proposed approach; the survey results and factors affecting software quality are reported in Section 4 and 5. Finally conclusions are driven in Section 6.

2. RELATED WORK

Surveys are greatly taken attention in the field of Software Engineering for the analysis of processes, methods and maintenance of software products (Hauge, 2007), (Li *et al*., 2008). The investigation of software products by means of surveys, comprising questionnaire and its purpose, can be carried out by number of ways such as online survey forms, face-to-face interviews, mails, emails and web blogs; the surveys are considered as very fast way to gather information. Research has been done for assessment of software Quality, identification of defects, reviews etc. in terms of state of art (Chandola, Banerjee and Kumar, 2009), (Czaja and Blair, 2005). The surveys, which focused practical consideration of the processes are conducted in (Ciolkowski, Laitenberger and Biffl, 2003), (Jedlitschka *et al*., 2007), (Scanniello *et al*., 2011).

The survey conducted in Ciolkowski *et al*. (Ciolkowski, Laitenberger and Biffl, 2003) analyzed state of practice; the major focus remained on the applications of walkthroughs, reviews and inspections. The authors highlighted the concerns regarding time consumptions and thus declared these techniques as inapplicable in practical usage. Jedlitschka *et al*. (Jedlitschka *et al*., 2007) analyzed practically adoption of inspections in decision making for the German software industries; their observation focused on technological impact on software Quality, cost, and development time.

Hauge (Hauge, 2007) investigated Open Source phenomenon and conducted a web-based survey, he observed 50% companies adopted open source coding in their software products. Torchiano *et al*. (Torchiano *et al*., 2008) reported 66% Italian companies experience migration tasks in his survey about analyzing state of practice of software migration. Scanniello *et al*. (Scanniello *et al*., 2011) conducted survey in Italian industries for the understanding the state of practice software Quality assessment and software defect identifications; they highlighted the software testing as popular technique and growing interest aims towards distributed inspection methods and deducted main problem as the lack of proper skilled employees. The aim of survey in Torchiano *et al*. (Torchiano *et al*., 2008) was to identify the major problems encountered for software Quality assessment and defect identification. However, they didn't highlight the factors degrading the software Quality during development lifecycle.

The intent of this paper is to identify the key factors and areas in the software development phases that affect software Quality. In particular, identification of the major factors causing degradation of software quality and making software quality a challenging issue during the software development lifecycle. Our work is carried out by conducting an online survey from development teams involved in the software development lifecycle in order to get insight into processes that are performed in the development lifecycle especially in local cities of Pakistan.

3. METHODOLOGY

The research mainly has been carried out performing the following steps: Research assumptions, Designing Questionnaire, Conducting Survey and Identifying Factors

The Research assumptions are made in the beginning of the survey in accordance with expected and well-known factors in literature and are described as follows:

- **R-1**: Non-standard procedures often result poor Quality
- **R-2**: Poorly gathered requirements and poor design, usually leads to compromise at Software Quality
- **R-3**: Unavailability of Quality Matrix agreements at the beginning of projects leads towards poor Quality
- **R-4**: Satisfied employees produce High Quality Software Applications
- **R-5**: Higher than 50% of total development in Testing phase yields Good Quality Software Products

The Questionnaire is carefully designed to satisfy our purpose of research covering the entire phases of Software Development and it also ensures the ethical requirements. In particular, Questions are included to figure out the major factors impacting Quality of Software at industrial level in Pakistan. The Questionnaire has been divided into sections as per various phases of development lifecycle such requirements, design, development and testing phases.

A survey has been conducted from Software Industries of big cities of Pakistan including Karachi and Lahore, which often contribute large in Software Market. In particular, Development teams (convenient contacts with authors) involved in Software Development are focused. We clarified the respondents about the research purposes and ensured them the data would be used solely for the research. A total of 47 responses (samples) were collected in a period of two months.

The identification of factors has been analyzed in the evaluation block in **(Fig.2)** based on domain knowledge, standards, and survey results. The survey results are also statistically verified with Research assumptions made earlier for finding factors. The Bird view block diagram of the methodology is illustrated in **(Fig.2)**. The findings of the work and survey results are described in the following sections in detail.

4. SURVEY RESULTS

Among the 47 respondents (samples), who participated in the survey, the city-wise distribution observed is as following: 38% respondents were from Islamabad, 26% from Lahore, 21% from Karachi, 13% from Hyderabad and rest 2% worked from home at various locations.

Regarding Qualification of respondents, only one (2%) had Higher Secondary Schooling, while 57% were with Bachelor's Degree, 40% with Master's Degree and None (0%) was found having PhD degree. Furthermore, only 9% (only 4) were found with female gender, the rest 91% (43 respondents) were male. The respondent's age ranged from 20-29 years accounted to about 85% and the remaining 15% were in the age-group ranging in between 30-40 years.

Regarding work experience, 19% of respondents were having more than 6 years of work experience, where as 9% of respondents had work experience ranging from 4-6 years. Rest 28% of respondents had 2-4 years of work experience and 45% had less than 2 years of work experience.

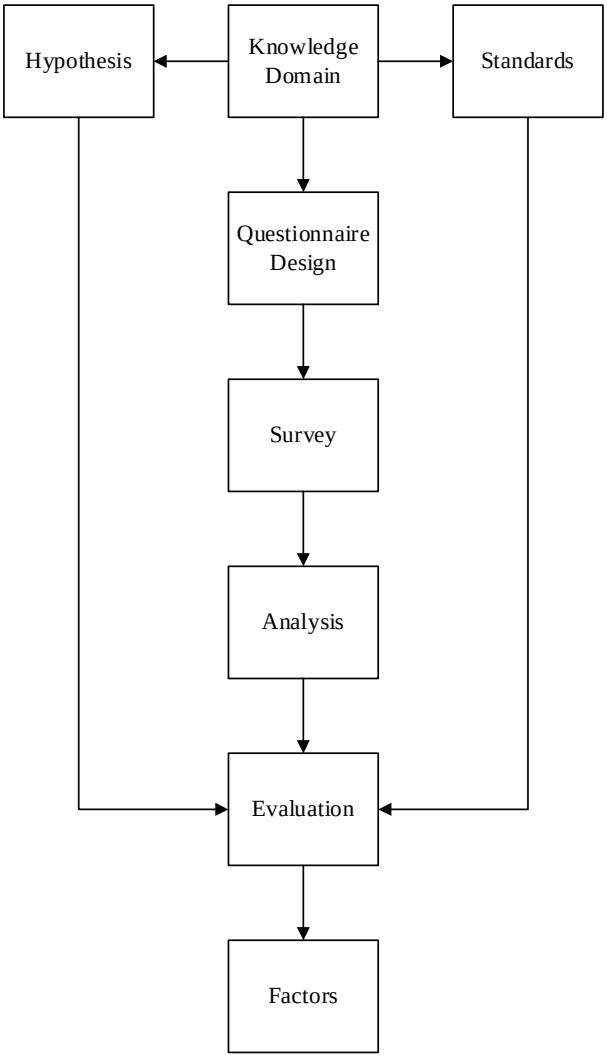

Fig.1 : Block diagram of Identification of Factors

The organizations having number of employees below 20 are considered as small, number of employees in between 20 and 99 are considered medium and over 100 are regarded as large organizations. We collected samples from organizations categorized as small, medium and large organizations by 21%, 40% and 38% respectively.

Regarding work environment and administrative behavior towards its employees, only 2-4% of respondents reported poor environment and/or poor behavior, whereas rest of them answered as Fair, Good and Excellent. Since, environment and administrative behavior maintain a key role in business domain, the survey results represent a higher number that is, 97% of respondents were happy with their organizations environment and administrative behavior. Moreover, 60% of the employees who took part in the survey, reported their colleagues as very much co-operative and 36% mentioned the conditional co-operation based on availability of leisure time.

The already made Research assumptions are analyzed in the light of survey results.

- **R-1**: Non-standard procedures often result poor Quality

To verify the Research assumption **R-1**, we asked the respondents whether they follow organizational policies, standards, and procedures for the requirement gathering and design phase. The **(Table 1)** reports the survey responses for both Requirement and Design Phase.

Table 1 : Requirement, Design standards and procedures

Phase Name	Yes	No	Don't Know	Total
Requirement	24	17	6	47
Design	22	18	7	47

The Quality of Software deliverables is measured according to the number of bugs reported by clients, once deliverables are handed over to the clients. Based on this measure, we computed the percentile of Good Quality and Poor Quality for the data described in **(Table 1)**. To validate the statistical significance of the Quality measure, we used t-test (Dietterich, 1998) at 5% significance level. The t-test produced p value 0.0002 that validates the quality measure statistically significant. Therefore, Research assumption **R-1** stands true regarding Quality of Software.

- **R-2**: Poorly gathered requirements and poor design, usually lead to compromise at Software Quality

The data collected in the requirement elicitation without standard procedures is regarded as poorly gathered, while the design of the requirements also needs a proper documentation to implement them. The design failing to clearly define requirement and not following any design standards is termed as poor design.

The requirements gathering phase is not followed by all the organizations considered in the survey. The **(Fig.3)** represents the percentage of responses describing the requirement phase being applied within the organizations.

The survey tells 38% of developers are provided requirements in written as well as diagrammatic format, where as 43% with written but only narrative format and 17% with verbal. These results explain the poor designing phase.

Moreover, the results describe 19% of respondents don't follow requirement phase and 9% don't know about it. Among these respondents who don't gather requirements properly, deliver 46% software products with more than 40% bugs. Likewise, the results show the poor design produces 50% of their software products having bugs more than 40%. Hence, the survey results are consistent with Research assumption **R-2**. The poor requirement gathering and poor design leads towards poor Software Quality.

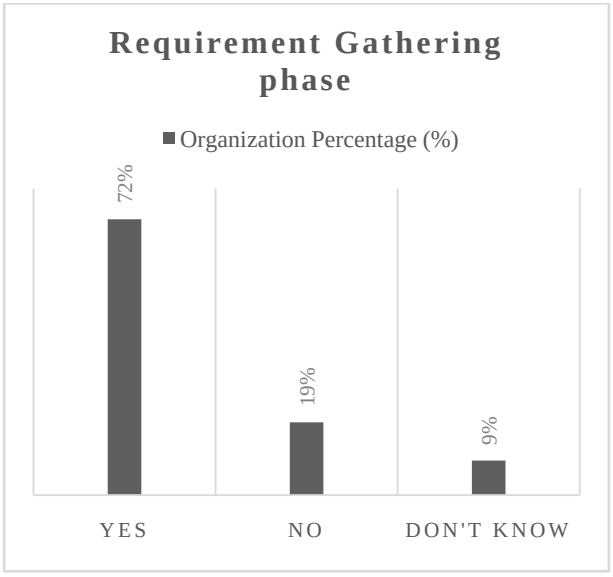

Fig.2 : Requirement Gathering Phase

- **R-3**: Unavailability of Quality Matrix agreements at the beginning of projects leads towards poor Quality

47% of respondents reported their organizations make agreements on Quality Matrix, while 38% responded in negation of it and 15% said they are not aware about it.

The 82% from 47% who made agreements say that less than 40% bugs are reported after delivery of software products. Unlike the other respondents who didn't make any agreements told that 66% projects are found having bugs with more than 40% once projects are delivered to clients. These figures of survey strongly verify the Research assumption **R-3**.

- **R-4**: Satisfied employees produce High Quality Software Applications

The **(Table 2)** describes the responses of satisfaction of employees with respect to their salary package and working timing.

We computed the responses of satisfied and unsatisfied respondents regarding their salary and work timing, to identify the impact on Quality of software products. The 43% of employees satisfied with work timing stated producing Good Quality (i.e. blew 20% bugs have been reported after its initial release) and 34% of satisfied employees regarding their salary package asserted Good Quality. On the other hand, below 20% of unsatisfied with respect to both salary and work timing told that the organizations where they work, produce good Quality products. We performed statistical testing (t-test) to validate the computed responses. The t-test resulted statistically significant the Research assumption **R-4**.

Table 2 : Employees satisfaction

Phase Name	Yes	No	Don't Know	Total
Salary	19	20	8	47
Work Timing	26	13	8	47

- **R-5**: Higher than 50% of total development in Testing Phase yields Good Quality Software Products

Among 47 samples, 37 (i.e. 79%) said they possess Quality Assurance Department within their organizations and only 10 (i.e. 12%) reported its absence. The **(Fig.4)** indicates the percentile of each city having QA department and also they declared the Good Quality products being developed since all these respondents reported projects having below 20% bugs after release.

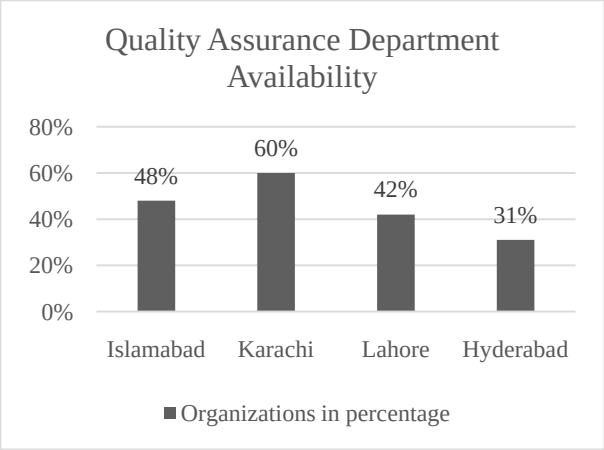

Fig.3 : QA department availability

While answering the Question about how much work is done on testing out of total development; the obtained responses are illustrated in the **(Fig.5)**. The organizations performing less than 40% of total development on testing are found having 52% projects with bugs. Similarly those organizations, which perform more than 40% of total development on testing, are producing Good Quality products since 50% of their projects are having below 20% bugs after release and 50% projects with less than 40% bugs; thus testing properly make projects successful and bug free. The survey results comply with Research assumption **R-5**.

Referring to **(Fig.5)**, only 4 out of 47 (~9%) respondents told that their organizations perform testing in range 61% - 80% of their total development work; similarly, 5 (~11%) perform in between 41% - 60%.

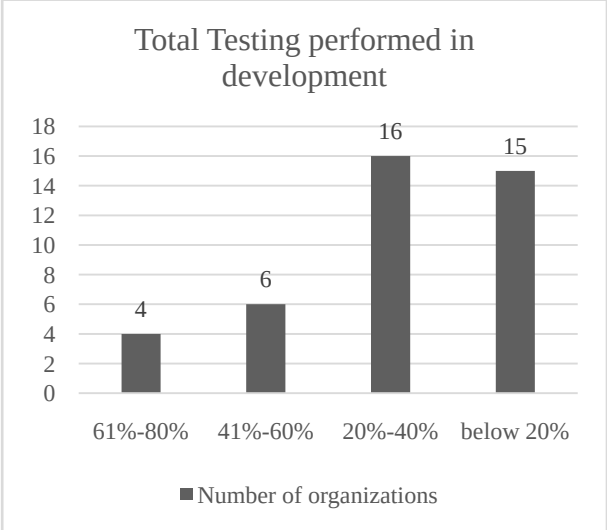

Fig.4 : Total Testing performed

5. FACTORS

In this section, we summarize the findings that cause software quality degradation according to survey results as well as domain knowledge. The Research assumptions are verified among larger portion of participants. Therefore, factors highlighted in the Research assumptions are strongly affecting the software Quality. We discuss the key factors in detail in the following.

5.1 Requirement Gathering and Design

Requirement gathering is one of the complex and difficult tasks in the software development; since, clients are more likely to change their demands. Moreover, if the requirements aren't properly defined to development team, it may cause delays and waste of time and effort. We found gathered requirement, being designed properly helps to produce good Quality products, and otherwise Quality is degraded.

5.2 Quality Matrix

The survey results indicated the role of Quality Matrix at the stage of requirement gathering helps

organizations to meet the desired specifications. Therefore, we recommend Quality Matrix as one of the key factor that affects software Quality.

5.3 SQA Plan

Indeed, Testing strengthens software Quality. The organizations, which apply QA Plan for the testing of software deliverables, often succeed. 98% respondents stated their QA team majorly performs testing. However, only 33% answered as applying Process Improvement. Survey showed the organizations having QA department, which following approved QA Plan from their QA team, produce better Quality software products in comparison with those organizations that don't have QA department.

5.4 Level of Standards Applied

Not only standard processes and procedures impact on overall Quality of software, but also the level of adoption of those processes within each phase of development. The results described 43% of the respondents said their organizations apply standard procedures at medium level (i.e. ignore some levels) and 21% said they apply always standards at each level. While 26% stated only a few levels standard processes are applied. We recommend the standard processes and procedures at each level of development lifecycle in order to get better software Quality.

Moreover, among the standards ISO (International organization for standardizations) standards were found being applied the most in the survey up to 38%, next IEEE (Institute of Electrical and Electronics Engineers) and CMMI (Capability Maturity Model Integration) having 23% responses both.

5.5 Project Delays

The 85% of the responses asserted the change of requirements during development phase that eventually causes delay in the projects. There are some other issues regarding delay in the project delivery including improper management, lack of planning for release schedule and overload of non-productive work load (i.e. unnecessary documentations). The faster development to release product deliverables on time, cause improper checking of functionalities and testing; since main goal is to release in time; hence such hurry in development compromise the Quality. We call project delays as a factor degrading the software quality.

The survey results indicated the role of Quality Matrix at the stage of requirement gathering helps organizations to meet the desired specifications. Therefore, we recommend Quality Matrix as one of the key factor that affects software Quality.

5.6 Employees Satisfaction

Interest in the work makes it worthy; it has been observed in the survey, employee satisfaction regarding their salary as well as work timing and environment makes them comfortable to produce better results. Instead unhappy employees are more likely to perform in poor manner. Our hypothesis **R-4** verifies our claim about employee satisfactions. In other words, employee satisfactions affect the software Quality.

5.7 Communication Medium

We also found that communication between development teams affects the overall development phases. 79% of the respondents replied e-mailing as the communication medium between teams. We recommend the physical meetings between team members in order to clearly understand the requirement specifications. The misunderstanding about the specifications, which are provided in emails, may cause waste of efforts and time. In our findings, communication medium between development teams plays an essential role develop better Quality products.

6. CONCLUSION

The survey reported in this paper aims at studying and identifying the major factors affecting software Quality during its development phases. The target populations participated in the survey includes development teams of major cities of software industries situated in Pakistan. The findings of the work, being the limitation of the research, indicate the quality indicative parameters for the Pakistani Software Industrial Market.

Larger portion of participants verifies the Research assumptions made in the study. The major findings, in our work, describe requirement gathering, designing of the requirements, agreements of Quality Matrix at the time of project contracts as one of the key issues, which are significantly important for the better Quality software deliverables. Furthermore, during development standard processes and procedures, proper planning and testing, adoption of standards in development life cycle, and communication between team members are also major concerns for the good Quality achievements. The work also deducted satisfaction of employees as one of the key factor for the better results.

The findings are interesting and can potentially support the software industries to manage the areas discussed in our investigation.

7. ACKNOWLEDGMENT

The authors are grateful to all participants who took part in the survey, and especially to Engr. Babar Tareen for his fruitful suggestions in Questionnaire Designing. Boehm,

REFERENCES:

Beck, K., M. Beedle, A. van Bennekum, A. Cockburn, W. Cunningham, M. Fowler, J. Grenning, J. Highsmith, A. Hunt, R. Jeffries, J. Kern, B. Marick, R.C. Martin, S. Mellor, K. Schwaber, J. Sutherland, and D. Thomas, (2001) *Manifesto for Agile Software Development*, Available: http://agilemanifesto.org/ [Accessed: 2015-04-12].

B.W. (1988) 'A Spiral Model of Software Development and Enhancement', *Computer*, vol. 21, no. 5,. 61-72, Available: http://dx.doi.org/10.1109/2.59.

Chandola, V., A. Banerjee, and V. Kumar, (2009) 'Anomaly Detection: A Survey', *ACM Computing Survey*, vol. 41, no. 3, July, pp. 15:1-15:58, Available: http://doi.acm.org/10.1145/1541880.1541882 .

Ciolkowski, M., O. Laitenberger, and S. Biffl, (2003) 'Software Reviews: The State of the Practice', *IEEE Software*, vol. 20, no. 6, November, pp. 46-51, Available: http://dx.doi.org/10.1109/MS.2003.1241366.

Czaja, R. and J. Blair, (2005) *Designing Surveys: A Guide to Decisions and Procedures*, 2nd edition, London: SAGE Publications.

Dietterich, T.G. (1998) 'Approximate Statistical Tests for Comparing Supervised Classification Learning Algorithms', *Neural computation*, vol. 10, 7, pp. 1895-1923, Available: DOI: 10.1162/089976698300017197.

Hauge, Q (2007) *Open Source Software in Software Intensive Industry - A Survey*, NTNU Innovation and Creativity,Available: http://www.idi.ntnu.no/grupper/su/su-diploma-2007/dipl07-hauge.pdf [Accessed: 30-6-2015].

Jacobson, I., G. Booch, and J. Rumbaugh, (1998) *The Unified Software Development Process*, 1st edition, Addison-Wesley.

Jedlitschka, A., M. Ciolkowski, C. Denger, B. Freimut, and A. Schlichting, (2007) 'Relevant Information Sources for Successful Technology Transfer: A Survey Using Inspections As an Example', Proceedings of the First International Symposium on Empirical Software Engineering and Measurement, Washington, DC, USA, 31-40.

Li, J., O.P.N. Slyngstad, M. Torchiano, M. Morisio, and C. Bunse, (2008) 'A State-of-the-Practice Survey of Risk Management in Development with Off-the-Shelf Software Components', *IEEE Transactions on Software Engineering*, vol. 34, no. 2, March, 271-286, Available: ISSN: 0098-5589 DOI: 10.1109/TSE.2008.14.

Pressman, R. S. (2001) *Software Engineering: A Practitioner's Approach*, 5th edition, McGraw Hill.

Royce, W. W. (1987) 'Managing the Development of Large Software Systems: Concepts and Techniques', Proceedings of the 9th International Conference on Software Engineering, Los Alamitos, CA, USA, 328-338.

Scanniello, G., F. Fasano, A. D. Lucia, and G. Tortora, (2011) 'Software Quality Assessment and Error/Defect Identification in the Italian Industry: Preliminary Results from a State of the Practice Survey', ICSEA 2011 : The Sixth International Conference on Software Engineering Advances, 589-594.

Sommerville, I. (2000) *Software Engineering*, 6th edition, Pearson Education Asia.

Torchiano, M., M. Di Penta, F. Ricca, De A. Lucia, and F. Lanubile, (2008) 'Software Migration Projects in Italian industry: Preliminary results from a state of the practice survey', 23rd IEEE/ACM International Conference on Automated Software Engineering - ASE Workshops 2008., 35-42.

Whitten, J. L., L.D. Bentley, and K. C. Dittman, (2003) *Systems Analysis and Design Methods*, 6th edition, Mcgraw-Hill.