

Requirements Validation Techniques: An Empirical Study

Hafiz Anas Bilal
Dept. of Computer
Science
COMSATS Institute of
Information Technology,
Islamabad, Pakistan

Muhammad Ilyas
Dept. of Computer
Science & IT
University of Sargodha
Sargodha, Pakistan

Qandeel Tariq
Dept. of Management
Sciences
COMSATS Institute of
Information Technology,
Islamabad, Pakistan

Muhammad
Hummayun
Department of
Computer Science & IT
University of Sargodha,
Sargodha, Pakistan

ABSTRACT

Requirements validation makes sure that the requirements written in software requirements specification (SRS) must be complete and consistent and are according to the customer's needs. It ensures the validity of user requirements by eliminating ambiguities and inconsistencies from SRS. Several techniques for requirements validation have been discussed in the literature. This paper gives an overview of requirements validation techniques which have been practicing in industry, which includes requirements inspections, requirements prototyping, requirements testing and viewpoint-oriented requirements validation. This paper also highlights pros and cons of these techniques. In requirements testing, special attention is given to TCD inspections.

General Terms

Requirements engineering, requirements validation, requirements elicitation

Keywords

Requirements validation, validation techniques, requirements engineering

1. INTRODUCTION

Requirements engineering is a process in which various activities are performed in a structured way to elicit, validate, and maintain user requirements [1]. These activities comprise of requirements elicitation, negotiation & requirements analysis, requirements specifications, requirements validation and requirements management.

During the requirements elicitation phase, the requirements engineers gather and elicit all the user requirements from different sources. These sources include domain knowledge, existing documents or systems, interviews or market surveys [1]. Most common way of eliciting requirements from the user is to interview them.

In the analysis phase, requirements engineers or system analysts analyze all the user requirements and negotiate with the customers to finalize the system requirements upon which all the stakeholder agree [1]. All the conflicting requirement are resolved in this process, and the budget and time constraints to fulfill those requirements [1].

Requirements Validation facilitates in resolving the conflicts between different stakeholders of the system due to incompleteness or any incompatibility of requirements within the available financial constraints [1].

The settled requirements are then base-lined and are specified

in a document called requirements specification document.

The requirements of a system always subject to change during the software development due to many reasons, which include a frequent change in customers' requirements, any change in the environment, any change in the legal requirements or constraints imposed by regulatory authorities, and change in business goals [1]. This frequent change in users' requirements is also called "requirements creeping". Requirements management helps in managing changing users' requirements. Any request to change requirement(s) which have been agreed upon earlier is made through a proper channel, called requirements change management [1].

Requirements validation makes it possible to eradicate all the conflicts among requirements in software requirements specification [1]. It relates to requirements analysis since it is more concerned with detecting problems with the requirements, i.e. inconsistencies, incompleteness and ambiguities among requirements. It highlights the ambiguous requirements - which may have multiple perceptions, inconsistent requirements - in which the same requirement is stated differently at different places in SRS, and incomplete requirements - which do not convey a complete meaningful requirement.

Requirements validation is crucial because the errors identified in the later phases of the software development would be more expensive to rectify because the cost of fixing a problem which arises due to the problem in requirements is much more expensive than fixing a design or code errors. Errors in requirements lead to failure of meeting customers' need and hence leads to failure of the project. Any change in requirements at the later phase means that you will need to make changes in design and architecture or implementation.

Errors in an SRS could lead to a huge rework cost when they occur in development phase or after the system is being deployed. Fixing an error at later stages of life cycle, i.e. design or development would require more cost. Because at this stage, more artifacts need to be changed. Moreover, the system needs to be re-tested after making those changes.

According to [3], the majority of the bugs, i.e. 56%, in a software project are due to the faulty requirements as shown in Figure 1. Almost half of the bugs are attributed to incompleteness and ambiguity of requirements and the remaining are due to omitted requirements. Since few deliverables are to be delivered at requirements engineering phase, the cost of resolving a bug at requirements engineering is lesser at this phase.

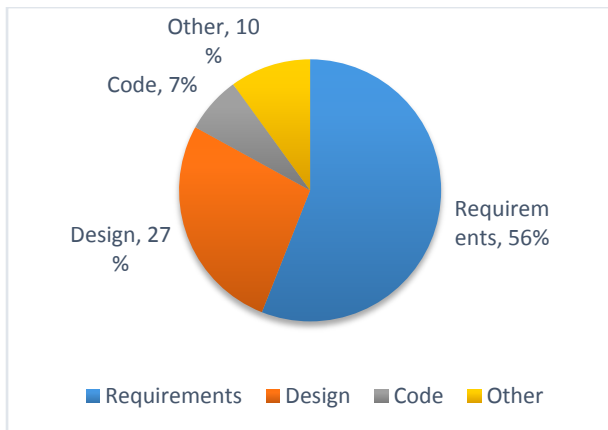


Fig.1: Distribution of Bugs in Software Projects

At the requirements validation phase, different types of checks are to be performed on the requirements in the software requirements specification document. These checks include:

a) Validity Checks

A user might consider the need of a system to perform certain functionalities. However, further analysis might highlight some other functions that are needed. Systems have diverse stakeholders with varying concerns with the system.

b) Consistency Checks

There should not be different meanings of any single requirement stated at different places/contexts in SRS. That is, there should not be contradictory constraints or dissimilar descriptions of the same system function at two different locations.

c) Completeness Checks

The requirements ought to be complete and it includes all the requirements that define all functions and their respective constraints.

d) Realism Checks

It must be ensured using existing technology that the requirements can actually be implemented. Schedule and budget constraints for the system development are also to be considered.

e) Verifiability

System requirements must always be specified in a fashion that they are verifiable in order to minimize the possibility for disagreement between customer and contractor

2. RESEARCH METHODOLOGY

A large portion of the research conducted on software engineering is based on exploratory studies which are carried out by the exploration of a specific problem domain or situation. These studies have been proven to be advantageous for defining the already done research [16]. These studies could be accomplished through literature reviews, taking responses via questionnaires or by conducting interviews [16]. During this study, our goal is to make a comparison and analyzing different requirements validation techniques which have been proposed in literature so far. Therefore, literature review would be a more appropriate methodology to accomplish this goal. Several research articles were collected and studied from IEEE, ACM, and Springer.

Following research questions will be discussed during this study:

- Identifying the difficulties which arise during requirements validation process?
- What requirements validation techniques have proposed in the literature so far?
- Identifying the pros and cons associated with those techniques?
- Identifying the significance of selecting a specific requirements validation technique in software projects?

3. CHALLENGES TO REQUIREMENTS VALIDATION

The application of an approach or a framework for requirements validation is a challenging task. Many organizations perform requirements validation in “ad-hoc” fashion due to the dearth of capable technical staff or appropriate training or knowledge & expertise of these techniques. Development teams put their most focus on testing phase.

4. REQUIREMENTS VALIDATION TECHNIQUES

The goal of using Requirement validation techniques is to guarantee that specified user specifications have been stated completely and there is no faulty requirement in the SRS document. Several techniques for requirements validation are practiced in industry, which includes, requirements prototyping, requirements reviews, viewpoint-oriented requirements validation and use-case based modeling [1]. In this paper, some of those techniques have been discussed.

4.1 Inspections

Fagan 1976, presented the concept of inspections, and they have been using it as a dominant technique for detecting errors [10]. Literature shows that inspections can identify 50-90 % of defects [9]. Inspections are a mean of confirming work-products by manually examining the product. It is carried out by a small group of peers to ensure that it is correct and conforms to the requirements and specifications of the product. ISO / IEC 15504 and CMMI have also recommended inspections as a part of requirements validation process [11, 12].

4.1.1 Phases of Inspection

Several steps are included in the inspection process, this section will describe these in detail.

a. Planning

Planning phase focuses on conducting a meeting for inspection process when any work-product(s) for inspections are determined. Document to be reviewed is identified and agenda for the inspection is set. Participants are identified who will attend the inspection meeting and also the schedule is set. Exit and entry criteria for the inspection meeting are identified [4].

b. Overview

During overview, the author(s) of the SRS read(s) the SRS being inspected loudly as the purpose of this process is to make the SRS under inspection easier to comprehend for participants of inspection. A few limitations are also linked with this phase [4]. Inspection meetings require extensive time and more effort. Moreover, the independent evaluation of SRS being inspected is not guaranteed. For instance, it might be possible that in the overview, all of the inspectors put their focus on a particular part of (SRS) and it might take more

time to reach any conclusion.

Glib and Graham named review process as “Kickoff Meeting” and classified this process as discretionary process [4]. With the help of it, authors could be able to describe essential aspects of the SRS to the inspection team.

c. Defect Detection

The essence of the inspection process is detecting defects [4]. It is performed to identify the defects in the SRS. This phase could be performed in either of two fashions. Each participant inspects the SRS and identifies/detects defects in it or a group meeting where more than one inspectors identify defects in the SRS.

In [4], authors gave the idea of performing the inspection in phases and named it “Phased Inspection Method”. In which, all inspection phases are further categorized or sub-divided into sub-inspections and same teams with their responsibilities are assigned to each sub-inspection. [4].

Porter et.al [4] has shown empirically that sub-inspections could lead to an increase in cost [4]. In a sub-inspection process, when a defect is detected it is repaired and then again it goes through the same inspection process. This increases the cost of the project.

d. Defect Correction

The identified defects in SRS at the defect detection stage are then rectified in this phase and this is done by the author of the software requirements specification (SRS). The author of SRS is responsible for the removal of all the issues detected in SRS in inspection meeting. The purpose of this phase is to ensure that the identified requirement defects have been eliminated from the (SRS).

e. Follow-up

In order to make sure that required modifications in SRS have been made accordingly, the moderator and author are responsible for this. The goal of the follow-up process is to make sure that author(s) of the SRS have rectified all the stated incomplete or/and inconsistent requirements or the detected defects [4]. If a significant amount i.e. 10% or more, of SRS, is modified, then further inspection might be conducted.

4.1.2 Roles in Inspection

In order to perform the inspection smoothly and to qualify it as a true inspection, the process has to follow specified process and all the participants of the inspection play their well-defined roles. Inspection team normally consists of 3 to 8 participants with the following roles [5].

a) Moderator

The moderator is the vital participant of the inspection process as it leads the inspection process. Its responsibilities are to schedule inspection meeting, controlling the meeting, reporting results of the inspection and to follow up rework issues. The moderator must be trained in a way so that he/she has expertise in conducting inspections, selecting teams with strong technical skills. [5].

b) Author

It is the person who created the SRS being inspected. He is responsible for answering the questions raised related to SRS in the meeting [5]. The author can't serve as reader, recorder or moderator [6].

c) Reader

The reader is responsible for presenting the inspected SRS and leads the inspection team [6]. He narrates the sections of

the SRS as they proceed through the meeting. The reader paraphrases the work products to the participants, i.e. explaining what a particular section of the design of code is supposed to do.

d) Recorder (Optional)

All the issues or defects identified in the SRS and then recorded are classified by the recorder. The recorder also prepares an issue list and also identify who will be responsible for resolving those issues [5].

e) Inspector

Inspector's responsibilities are to find defects in the product. Actually, all the participants play the role of inspectors as an additional responsibility. All these participants identify all the possible defects in the SRS and after resolving those defects, re-inspection of the work products can be performed if needed.

4.2 Requirements Prototyping

Requirements prototyping is a fundamental technique for validating users' requirements as it represents the shell of an actual system to be built. Prototypes facilitate in validating requirements by providing a valuable insight into the system. Prototypes are a good tool for validating requirements especially when you are not pretty confident that you have a good set of requirements. Two types of prototypes have been discussed in the literature, i.e. throw-away prototypes, and evolutionary prototypes.

“Throwaway prototypes” aids in identifying the requirements that were not properly understood. Throwaway prototypes are discarded or thrown away after the user feedback once the initial set requirements are built in the prototype. It helps in resolving requirements conflicts between the development team and the customers by taking feedback on the prototype. If both are agreed upon a defined set of requirements, then the prototype is discarded and the requirements are assimilated in SRS [8].

“Evolutionary prototyping” on the other hand, is performed on a set of settled requirements and is subject to quality constraints as imposed in the software development [8]. Evolutionary prototypes are built from initial requirements and gradually refinement is made with depending upon feedback from the user.

Following are few advantages and disadvantages of prototypes:

- Prototypes help the customers to get a visual insight of the developing software system. With the help of prototypes, it is easier for them to identify issues with requirements and define the need for any additional requirements if they are not available in the current prototype.

Some of the drawbacks of the prototypes are listed below.

- It takes more time to prepare prototypes and hence results in more cost.
- If the paper prototypes are approved by the customer, they could not be converted into executable versions after validation.

4.3 Requirements Testing

The aim of executing requirements testing is to ensure the validation of the requirements in the (SRS) instead of validating them in software system [7]. For this purpose, test cases are generated for all the stated requirements, time for

writing test manuals and/or economic resources. The cost incurred in this process is included in the cost of requirements validation [1]. Requirements testing aids in the identification of ambiguous or incomplete requirements in a way that if there some problem occurs while executing a test case for any specific requirement, it gives an indication that there is some issue with that requirement [7].

The test cases prepared for testing the requirements could also be used for the testing of the full system in a later phase of testing of system development [13]. The process of requirements testing is shown in Figure 2. Test case based inspections are referred as “TCD inspections - Test-case driven inspection” [13].

The concept of test case based inspection of the software requirements was proposed by Tony Gorschek and Nina Fogelstrom. In this technique, test cases are written for testing the given requirements of system and tester also involved in this testing process [13]. TCD Inspection process used is shown in Figure 3.

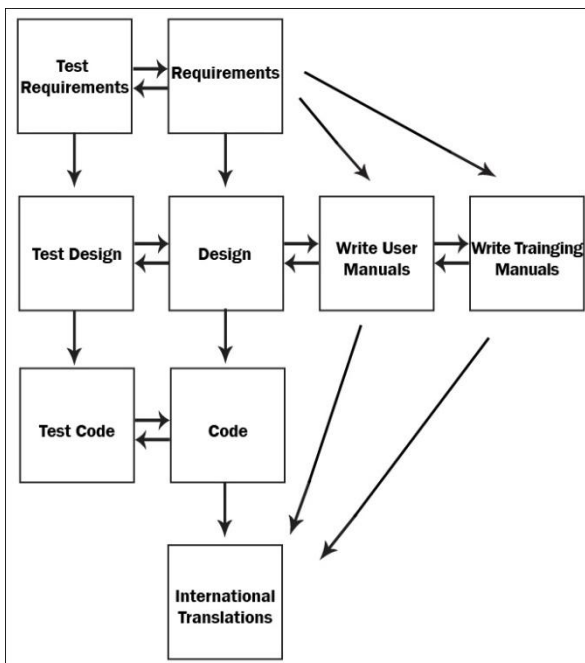


Fig 2: Requirements Testing Process

At step 1 of the TCD inspection process, Project Manager (PM) selects which user requirements are to be incorporated into the initial preparation of SRS or specification [13]. At step 1, PM chooses and makes a review of those requirements which are to be incorporated in the initial draft of SRS or specification. There can be multiple sources of requirements and it is not necessary to include all the requirements at this step. Most of the time this process is performed on an ad-hoc basis and the PM utilizes his personal knowledge and experience and can ask for a suggestion from his/her peers. The specified user requirements are then moved to the step 2. A proper specification template is used for specifying these requirements containing attributes such as ID, title, description etc. The output of this step is the formulated/specified requirements which will be the input for step 2.

At step 2, test-cases are generated and TCD inspections are performed for the requirements specified in SRS [13]. Because of the refinements of the requirements, some of the requirements are discarded or postponed. This will help in

removing inappropriate requirements or having low priority.

At step 3, priority is set for the inspected requirements according to the required criteria and project planning tasks are also executed [13]. Some requirements might also be discarded at this step.

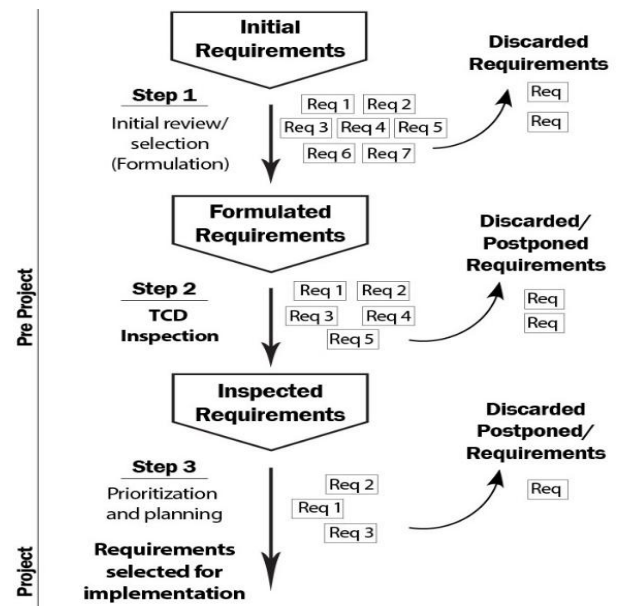


Fig 3: TCD Based Inspection

Following are the few advantages and disadvantages of prototypes:

- Unnecessary requirements could be eliminated with the help of requirements testing and the test cases generated in it could also be used later for the complete testing of software.
- It seems to be beneficial to large software organizations which have skilled and experienced testing teams. Organizations having experience of using this method can also offer training services to other organizations.
- The negative aspect of requirements testing is that it is more costly in terms of time spent on test case generation and then requirements testing using those test cases. It might not be useful for small organizations. Requirements testing require skilled requirements engineers and testers .Small organizations might lack experienced persons. Besides this, small organizations may not offer requirements testing training to the new resources.

4.4 Viewpoint-oriented Requirements Validation

Researchers have been using the concept that more sources of the information give a better understanding of a subject for many decades. Different sources and witnesses may provide conflicting and complementary recollections. This aids in ensuring the completeness and correctness issues in the requirements. In order to get benefit from this principle, different views must be compared and then analyzed in a systematic fashion (Freeman & Leite, 1991). The main goal of this process is to facilitate elicitation process. Viewpoint oriented validation performs this by comparing several different views and then providing a mechanism for the negotiation of different conflicts.

4.4.1 Definitions and Terminologies

Viewpoint oriented validation technique provides validation of requirements of a complex system in an early phase of the requirements elicitation. Viewpoint resolution identifies differences among different viewpoints, performs their classification and evaluation of those differences and integration of different alternative solutions into a single illustration (Freeman & Leite, 1991).

Some terminologies and their definitions used in this validation technique are described below:

The Universe of Discourse: It is defined as “the complete context in which a software would be developing, and it comprises of all the sources of information and all the stakeholders involved” [10].

Actors: They are the participants involved in the universe of discourse [10]. They could be classified into two categories: [10]. “Users on demand side” indicates customers and “Users on supply side” indicates requirements engineers, PM, or software development team.

Viewpoint: “It is the mental position and/or standing of any specific actor while observing/examining the universe of discourse” [10].

Perspective: “It is the group of the facts used for observing and modeling according to a specific modeling aspect and/or a viewpoint” [10].

View: The integration of different perspectives is termed as a View [10].

VWPL: Viewpoints are represented using viewpoint language [1]. More than one analysts, each having their own viewpoint participate in this validation (viewpoints) and the universe of discourse is modeled using VWPL.

Hierarchies: Authors in [10] classified the hierarchies that are used in the universe of discourse in two categories, i.e. “is-a hierarchy of concepts” and “parts-of hierarchy of concepts” [10].

The universe of discourse is modeled by analyst 1 and analyst 2, using three different perspectives (data, process, and actor) the previously defined hierarchies. A list of differences and their types are produced by analyzing perspectives and hierarchies [10].

The two perspectives are then integrated into views shown as view1 and view2 in Figure 4. The two views are then compared for verifying correctness and completeness. In figure 4, Oval shape represents the processes and Boxes represent the inputs and outputs of those processes [1].

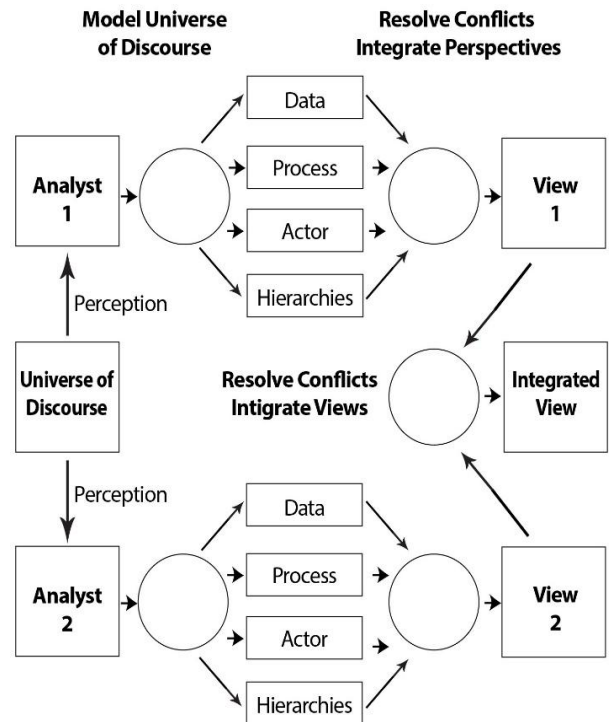


Fig 4: Viewpoint-Oriented Validation

Following are the pros and cons of viewpoint-oriented validation:

- Conflicting requirements can be easily identified on the basis of viewpoints.
- The main benefit of this technique is that it consists of views from different analysts and different perspectives.
- Limited work has been done so far on this validation process, this may cause some issues for requirements engineers to use this.

5. CONCLUSIONS

Requirements validation is a fundamental activity of requirements engineering process. Different validation techniques have been discussed in literature along with their pros and cons. In order to complete the software projects successfully without correctness and completeness problems in requirements, it is vital for software organizations to adopt some strategy for performing validation of requirements. This approach ought to clearly specify requirements validation technique(s) to implement. This will help them to minimize issues in requirements, i.e. conflicting requirements, ambiguities in requirements or inconsistencies etc. Requirements validation techniques have the significance that they supports defect detection that could be more problematic in later phases of SDLC. Organizations can save their cost of fixing bugs if they successfully detect all the bugs in requirements phase, and the project will also be deployed on time. If most of the bugs are detected and eliminated during requirements phase, then a little effort will have to be spent in the testing phase of regression testing and the hence cost will be low. A comparison of the techniques discussed in the paper is shown in Table 1.

Requirements prototyping involves collaboration with customers and takes their feedback after developing prototypes. This technique helps customers to have a visual insight of the system. Viewpoint-oriented requirements

validation allow the requirements engineers to identify conflicting requirements based on various stakeholders' viewpoints. These requirements validation techniques would facilitate to detect errors before they are being passed to other stages of SDLC.

Table 1: Comparison or requirements validation techniques

	Requirements Inspection	Requirements Prototyping	Requirements Testing	Viewpoint -Oriented Requirements
Team Size	Large	Small	Large	Small
Cost	More Costly	Less Costly	More Costly	Less Costly
Organization Size	Large	Small & Large	Large	Small & Large
Reuse	N/A	Yes	Yes	N/A
Customer Involvement	No	Yes	No	Yes

6. REFERENCES

- [1] Kotonya, G, Sommerville, I, *Requirements Engineering*, John Wiley & Sons, New York, 1998.
- [2] Thayer, R. & Dorfman, M. *Software Requirements Engineering 2nd Edition*, IEEE Computer Society Press, California, 1997.
- [3] Gary E. Mogyorodi, "What are Requirements- Based Testing", *CROSS TALK The Journal of Defence Software Engineering*, March 2003.
- [4] Laitenberger O, "A Survey of Software Inspection Technologies", *Handbook on Software Engineering and Knowledge Engineering*, Citeseer, 2002.
- [5] Steven R. R., *Software Verification and Validation for Practitioners and Managers*, Artech House Publishers, 2nd edition (August 1, 2001).
- [6] Ackerman, A. F., Buchwald, L. S., and Lewsky, F. H., "Software Inspections: An Effective Verification Process", *IEEE Software*, 1989, pp. 31-36.
- [7] Sommerville I, Pete Sawyer, *Requirements Engineering*, John Wiley & Sons, New York, 1997.
- [8] Siddiqi J, I. Morrey, R. Hibberd, G. Buckberry, "Towards a System for the Construction, Clarification, Discovery and Formalisation of Requirements", *proceedings of first international conference on Requirements Engineering*, IEEE, 1994, pp.230-238
- [9] Laitenberger O, Beil T, Schwinn T, "An industrial case study to examine a non-traditional inspection implementation for requirements specifications," *presented at Proceedings of the Eighth IEEE Symposium on Software Metrics*, Los Alamitos CA, 2002.
- [10] Leite J.C.S.P, Freeman P.A, "Requirements validation through viewpoint resolution", *IEEE Transactions on Software Engineering*, Vol 17, 1991, pp.1253-1269
- [11] CMMI-PDT, "Capability Maturity Model Integration (CMMI), Version 1.1," in *CMMI for Systems Engineering, Software Engineering, Integrated Product and Process Development, and Supplier Sourcing Version 1.1 (CMMI- SE/SW/IPPD/SS, V1.1)*. Pittsburgh, 2002
- [12] ISO/IEC, "Software Process Assessment TR 15504:1998," vol. 2004. <http://www.sei.cmu.edu/iso-15504/>: ISO/IEC, 1998.
- [13] Gorschek T, Dzamashvili N - Fogelstrom, "Test- case Driven Inspection of Pre-project Requirements-Process Proposal and Industry Experience Report", *in proceedings of the Requirements Engineering Decision Support Workshop held in conjunction with the 13th IEEE International Conference on Requirements Engineering*, 2005.
- [14] Briand L.C, Emam K.E, Freimut B.G, Laitenberger O , "A comprehensive evaluation of capture-recapture models for estimating software defect content", *IEEE Transactions on Software Engineering*, Vol 26, No 6, 2000, pp. 518-5
- [15] Fagan M.E, "Design and Code Inspection to Reduce Errors in Program Development," *IBM Systems Journal*, vol. 15, 1976, pp. 182-211.
- [16] Christian W.D, *Projects in Computing and Information Systems*, Pearson, 2005, ISBN 0321263553.
- [17] Carolyn B. S, "Qualitative Methods in Empirical Studies of Software Engineering", *IEEE transactions on Software Engineering*, Vol 25, No 4, July/August 1999