The Focus on Usability in Testing Practices in Industry

Marta Kristin Larusdottir, Emma Run Bjarnadottir, Jan Gulliksen

HAL Id: hal-01055489
https://hal.inria.fr/hal-01055489

Submitted on 12 Aug 2014

The Focus on Usability in Testing Practices in Industry

Marta Kristin Larusdottir, Emma Run Bjarnadottir and Jan Gulliksen
[1]Reykjavik University
Menntavegur 1
102 Reykjavik, Iceland
marta@ru.is
[1,3] The Royal Institute of Technology
SE-100 44 Stockholm, Sweden
gulliksen@kth.se
[2]Calidris Ltd.
Vesturhlid 7
105 Reykjavik, Iceland
emmarun1@gmail.com

Abstract. A study exploring the focus on usability in testing practices in software development teams in Iceland using the agile software process Scrum is described in this paper. A survey was conducted to describe how testing is conducted and to what extent testing techniques are used. The results show that unit, integration, system and acceptance testing are the most frequent testing techniques used, but usability testing is not that common. Neither are alpha, beta, performance/load and security testing. Interviews were conducted to exemplify how practitioners conduct usability testing and what they describe as the difference between usability and acceptance testing. Some examples from the interviews show that practitioners are willing to do formal usability testing on extensive parts of the system, but because the iterations in Scrum are short and the changes to the system in each iteration are small, formal usability testing does not fit into the project work.

Keywords: Usability, software testing, agile development, Scrum, practitioners.

1   Introduction

This paper describes and discusses the results of a study conducted in April though August 2009, on how usability is emphasized in the practices of software testing in software development teams in Iceland using the agile development process Scrum [1]. A survey was conducted to study how usability is emphasized in testing by development teams in comparison to other testing techniques. Furthermore the team members were asked to compare their experience of testing in Scrum to their experience of testing in any prior development process. Interviews were used to exemplify how practitioners conduct usability testing and especially what they describe as the difference between usability and acceptance testing.

The need for studying the complexity of usability evaluation in practice was stated by Wixon [2], where he suggests that the usability evaluation should be studied in its real context, not on simulated systems or hypothetical models. Many researchers share

his opinion, for example Cockton and Woolrych [3] recently stated that knowledge from empirical studies is needed. They suggest that these studies could use questionnaires, interviews or controlled experiments as research methods.

Furthermore the need for empirical studies of testing practices and evaluation has been identified, to develop an understanding of the organizational rationale for how software testing is practiced, for example in [4] and [5]. Software testing research has focused on extending the quality of testing in technical terms, like improving the design of tests, designing tools to support testing and measuring test coverage and efficacy of fault finding techniques [6]. Fault removal is only one of the potential goals for software testing; other goals include the evaluation of specified quality, e.g. usability or performance testing. It is preferable to spend the test budget to apply a combination of different techniques [6], even if one technique is shown to be the most effective. The actual use of different testing practices and the combination of those in the software industry needs further observation.

Modern agile development processes and traditional processes are often perceived as being opposed [7]. Traditional approaches tend to emphasize specifications and verification. These processes focus on the abstract and on correctness. Agile processes emphasize test and rapid prototyping, with focus on the concrete and the convenient and on validation. Both approaches emphasize co-development of the program and its evaluation. More and better empirical studies of agile software development within a common research agenda is needed [8]. An interesting aspect is whether it is easier to process testing in agile development processes than traditional processes.

Scrum is one of the agile development software development processes [1] where the focal aspects are simplicity and speed [9]. It progresses via a series of iterations called sprints, typically 2 – 4 weeks long. At the end of each sprint, a potentially shippable product increment should exist. The key roles in Scrum are Product Owner that defines the features of the product and defines what should be in each sprint, the Scrum Master that represents management to the project and software team members [1]. In Scrum there are four main activities, the sprint planning, the sprint review, where the team present what was accomplished in the sprint, the sprint retrospective, where the whole team discusses what is and what is not working in the project work and finally the daily scrum meeting, where all the team members report on their tasks. There are three main artifacts in Scrum, the product backlog, containing all desired work in the project, the sprint backlog that contains what will be done in the particular sprint and the burndown charts that illustrate the progress in each sprint [1]. One of the key factors in Scrum is that the team members never get ahead of the testers, because a requirement is ideally not "done" until it has been tested [10].

The focus of the study described in this paper is to research the actual use of testing techniques used in the software industry where the Scrum process is used, particularly focusing on how usability testing is performed in comparison with other testing techniques.

The research questions explored in the study are:

1. How is testing practiced in Scrum projects in the industry?

2. To what extent is usability testing performed compared to other testing techniques?

3. How does usability testing differ from acceptance testing in Scrum projects?

2   Research Methods

There were two research methods used in this study, a survey and interviews. The survey was done by sending a web based questionnaire to practitioners in software development companies using Scrum as their development process in Iceland. The respondents were all asked to take part in interviews after the survey. Six interviews were conducted from the pool of 9 people that agreed to being interviewed.

2.1   The survey

The survey included 26 questions, 5 open questions and 21 multiple choice questions. There were 5 main sections in the questionnaire: a) background and experience of the respondent, b) information on the company where the respondent works, c) the software development process(es) used in the company, d) to which extent and who is conducting different testing techniques and e) the change in conducting software testing when compared to previous/parallel software development process.

The survey was sent to 20 software companies that had confirmed using the Scrum process and doing software testing. The questionnaire was distributed to a single contact at each company, previously chosen to fit the target audience. The respondents were asked to send the questionnaire to another person in the company, if they thought that person fitted better the target audience. The results from the survey are in most cases based on 25 responds from 18 companies. In some cases one answer per company is used, e.g. when asking about the company's industrial section.

Most of the respondents, 76%, do have a degree in computer science or engineering. The majority of the respondents were male 68%, 20% female and 12% gave no reply. Nobody had been working for less than a year, 28% had been working for 1 – 3 years, 24% for 4 to 9 years and 28% for 10 to 14 years. Sixteen percent had been working for more than 15 years and one respondent did not reply.

Half of the companies are in the computer service or software development industry, 22% are in the banking, insurance or finance, 11% are in gaming, 11% in telecommunication and telephone, and 6% in the airline industry. The results on the size of the software companies show that 33% had up to 19 employees, 28% had 20 to 59 employees, and 33% over 60 employees. No data was given in 6% of the cases.

All the 18 companies use Scrum as a development process to some extent. Almost half of the respondents say that 81% – 100% of the software development department use Scrum, 34% said it is 21% - 80% and 22% were using it to the extent of 0% – 20%. In 61% of the cases another software development process was also used beside Scrum within the company, for example the agile process XP, Waterfall process and RUP. Interestingly 44% mentioned using their own process besides other variations. The respondents were asked about their primary role in relation to the Scrum development process. Almost half of them 44%, were Scrum Masters, 24% software testers, 20% Product Owners and 12% had other role.

2.2   The Interviews

To strengthen the results from the survey and to exemplify the reason for the results, six interviews were conducted, three with software testers and three with Scrum Masters. One of the goals of the interviews was to gather data to deepen the understanding of the results for the third research question on the difference between usability and acceptance testing.

The same person conducted all the interviews face to face and prepared a set of questions for the interviews. Each interview was recorded and transcribed afterwards. The interviews were semi structured, so the interviewer asked more questions than in the interview guide, if some issues were relevant in that particular case.

3   Results

The results are described in three subchapters. First the results on how testing is practiced in Scrum in the companies involved are described. Second, the results on how usability testing is practices compared to other testing techniques are represented and finally the results on the difference between usability and acceptance testing are explained.

3.1   Results on How Testing is Practiced in Scrum

Table 1 describes the results on fundamental activities of the Scrum development process. Many of those activities are practiced up to 90% or more. The activity that is least practiced is playing planning poker, which should be used to estimate the effort needed to complete the requirements in the particular sprint. Potentially shippable product after each sprint has been one of the main selling points for Scrum. There are only 64% of the respondents that say that this is practiced.

Table 1. The practice of fundamental activities in Scrum development

Scrum activity	Percent of repondents
The role of Product Owner is used	88%
The role of Scrum Master is used	92%
Requirements/user stories are kept in a Product backlog	84%
Planning poker is played on Product backlog items	60%
A Sprint backlog is established at the beginning of a sprint	92%
Each sprint/iteration is 2 - 4 weeks long	96%
Potentially shippable product exists at the end of a sprint	64%
A sprint review meeting is held at the end of a sprint	80%
A sprint retrospective meeting is held at the end of a sprint	88%
Scrum (Agile) metrics are used, like burn down charts	76%

The respondents were asked to name at least three things they believed to be positive effects of the Scrum process and compare with parallel/prior development process. Eighty percent of the respondents gave some comments to this question. The comments indicate that the respondents are generally happy changing over to using Scrum as their development process. Some examples are "more productivity", "QA involvement is a lot better", "combined responsibility", "better morale", "relationship with client improved", "customer in closer connection with the development and can change functions before final release", "less disturbance", "easier to handle changes in requirements/priorities" and "stand up meeting keep communication open".

Respondents were also asked to comment on what could be improved in relation to Scrum. The results indicate that there is always a room for improvement. To give some examples, "don't over commit", "more design documentation", "internal marketing of Scrum projects" and "organizational changes" are all comments from the respondents.

Table 2. The applicability of testing practices in Scrum projects

Testing practice	Percent of repondents
Software testing falls within the frame of "done" in each sprint	64%
Software testing is squeezed into the end of each iteration	36%
Software testing is not well integrated with coding and ends up one sprint behind	20%
Software testing is performed in a separate test environment	44%
Good management of version control	60%
Before a major version release, there is a bug-fix sprint	40%
Software testing became easier than in a parallel/prior process	44%
Overall more software testing is done than in a parallel/prior process	44%
Overall less software testing is done than in a parallel/prior process	12%
Programmers started using more test-driven development/design	48%
Software testers became more involved throughout the whole development	72%

In Table 2 are results on how testing is practices in Scrum and how that compares to prior or parallel process. The positive results are that 60% say that they have good management of version control and 64% say that testing falls within the frame of "done", which means that the required functionality is not delivered to the customer before it has been thoroughly tested. The testing does only end up one sprint behind in 20% of the cases and is squeezed into the end of each sprint in 36% of the cases.

When asked about the testing practice in comparison to prior or parallel process, 72% of the respondents say that testers became more involved throughout the whole development which is positive. The result show that 44% of the respondents say that software testing became easier than in parallel or prior process. Also 44% of the respondents say that more testing is done and only 12% say that less testing is done. These results show that Scrum has positive effects on testing practices.

3.2 Results on Usability Testing Compared to other Testing Techniques

The results from the survey in this subchapter build up to answer the research question:

To what extent is usability testing performed in relation to other testing techniques?

The respondents were asked to answer how much testing is done of each testing technique, who is performing the testing and give reasons for why particular testing technique is used less than other testing techniques. The results are described below. When respondents were asked about the testing techniques, these were explained as described in table 3, according to descriptions in [11].

Table 3. The description of each testing technique that the respondents got in the survey.

Testing technique	Description
Unit/component testing	The testing of individual software components.
Integration testing	Testing performed to expose defects in the interfaces and in the interactions between integrated components or systems.
System testing	The process of testing an integrated system to verify that it meets specified requirements. This includes test design techniques like boundary valued analysis and is usually done by internal software testers.
Acceptance testing	Formal testing with respect to user needs, requirements and business processes conducted to determine whether or not a system satisfies the acceptance criteria and to enable the users, customers or other authorized entity to determine whether or not to accept the system.
Usability testing	Testing to determine the extent to which the software product is understood, easy to learn, easy to operate and attractive to the users under specified conditions.
Alpha testing	Simulated or actual operational testing by potential users/customers or an independent test team at the developer' site, but outside the development organization. Alpha testing is often employed for off-the-shelf software as a form of internal acceptance testing.
Beta testing	Operational testing by potential and/or existing users/customers at an external site not otherwise involved with the developers, to determine whether or not a component or system satisfies the user/customer needs and fits within the business processes. Beta testing is often employed as a form of external acceptance testing for off-the-shelf software in order to acquire feedback from the market.
Performance/load testing	The process of testing to determine the performance and/or measuring the behavior of a component or system with increasing load, e.g. the number of parallel users and/or numbers of transactions, to determine what load can be handled by the component or system.
Security testing	Testing to determine the security of the software product.

The participants were asked to rate how much testing is done in their Scrum projects. The results are shown in table 4. The most common answer is showed in bold letters and the results on usability testing are shown in italics.

Table 4. The extent to which the testing techniques are used in Scrum projects, N = 23.

Testing technique	Yes, a lot	Yes, some	So and so	Little	No, not at all
Unit/component testing	22%	**35%**	26%	13%	4%
Integration testing	17%	**35%**	31%	13%	4%
System testing	**39%**	30%	22%	9%	0%
Acceptance testing	30%	**44%**	13%	13%	0%
Usability testing	*4%*	*22%*	*35%*	*35%*	*4%*
Alpha testing	4%	13%	17%	17%	**48%**
Beta testing	9%	22%	9%	17%	**44%**
Performance/load testing	0%	26%	26%	**35%**	13%
Security testing	4%	22%	8%	**39%**	26%

The results in table 4 show that system testing is the most common testing technique. Fairly common are also acceptance, unit and integration testing. Usability testing and performance/load testing have a similar use pattern, though more respondents chose the alternative "so and so" for usability testing than for performance/load testing. Security testing is not done as much as performance/load testing, but still there are 26% that do security testing to some extent or a lot. Almost half of the respondents say that alpha and beta testing are not used at all.

Table 5. The person using each testing technique in Scrum projects.

Testing technique	Programmer	Software tester	External Software tester	Customer	Others	N
Unit/component testing	**73%**	17%	3%	7%	0%	23
Integration testing	**44%**	36%	6%	11%	3%	22
System testing	21%	**50%**	13%	11%	5%	23
Acceptance testing	6%	33%	8%	**44%**	8%	21
Usability testing	*9%*	*43%*	*14%*	*26%*	*9%*	*22*
Alpha testing	13%	**35%**	13%	30%	9%	12
Beta testing	9%	32%	14%	**41%**	4%	13
Performance/load testing	**47%**	43%	3%	3%	3%	21
Security testing	**44%**	33%	4%	4%	15%	18

In Table 5 results on who is using each testing technique are listed. Since some answers were lacking, the actual N is labeled for each testing technique in the table. Software testers use all the testing techniques extensively as would be expected, 50% of the system testing is done by software testers, 43% of the usability testing and 43% of the performance testing. Programmers are mainly responsible for unit/component testing. Customers are taking considerable part in acceptance, beta and alpha testing and usability testing is done by customers in 26% of the cases. Performance/load and security testing are done jointly by software testers and programmers.

Table 6. The reasons for using some testing techniques less than others

Testing technique	Lack of training/ knowledge	Lack of budget	Lack of time	Other	N/A	N
Unit/component testing	**36**%	0%	32%	5%	27%	22
Integration testing	11%	0%	42%	0%	**47**%	19
System testing	7%	0%	**47**%	0%	**47**%	15
Acceptance testing	7%	0%	27%	7%	**60**%	15
Usability testing	*20%*	*15%*	*35%*	*10%*	*20%*	*20*
Alpha testing	0%	11%	11%	10%	**68**%	19
Beta testing	0%	11%	17%	11%	**61**%	18
Performance/load testing	26%	11%	**32**%	0%	**32**%	19
Security testing	**47**%	5%	16%	0%	32%	19

Table 6 analyzes what causes some testing techniques to be used less than others. Again some answers were lacking, and the actual N is labeled for each testing technique. The main reason for why usability testing is not conducted is lack of time. Lack of training was the reason in 20% of the cases and lack of budget in 15% of the cases. In this particular question many of the respondents chose the alternative "not applicable", which probably means that they do not know the reason.

The respondents were finally asked if they were missing any kind of software testing. Only a few answers were given, mentioning exploratory and regression testing.

3.3   Some Examples for the Reasons for these Results

Six interviews were conducted to get some examples on the reasons why usability testing is not practiced to a wider extent, how the interviewees explain the importance of usability testing and if usability testing is something that could be ignored. Furthermore some examples on how practitioners explain the difference between acceptance testing and usability testing are described.

Three interviews were conducted with software testers and three with Scrum Masters. All the interviewees had answered the questionnaire three months earlier and the interviewer had their answers to refer to during the interview. The results are explained in the following.

3.3.1 The Importance of Usability Testing

When asked about the importance of usability testing all the respondents mentioned that if a project is big and many changes have been made, formal usability testing would definitely be important. One respondent explained that if the project is really agile, the changes are not that extensive each time and the importance of being quick to market is strong, so usability testing is really not needed, because a shippable product has been delivered and the customers can complain. Furthermore because the changes are small, extensive usability testing is not needed and is too expensive. "The main thing is to confess your fault and change quickly according to the customers complaints so you can be very quick in adjusting to their needs" one of the respondents remarked. This respondent explained that asking for usability testing was really the customer's responsibility.

Some respondents remarked that the users were not always willing to take part because they were busy doing their own work and did not want to be involved in software development and one respondent said that usability is fuzzy, hard to measure and usability requirements are always changing.

All the respondents were asked whether they would like to do more usability testing if they had time and money. All the respondents wanted that at least occasionally. They were also asked if usability testing was something that could be ignored. None of the respondents wanted to ignore it. One of the respondents specified that there is a need for a formal usability test of the system once a year or every second year, but during other testing usability issues are implicitly considered, "It is always on my mind" the respondent commented.

3.3.2 The Difference between Acceptance Testing and Usability Testing

All the respondents were asked about what the difference between acceptance testing and usability testing is. Many explained that acceptance testing is more structured that usability testing. During acceptance testing, the functionality of the new system is shown to the customer in predefined steps. The customer has to sign that he or she has accepted that the requirements are fulfilled.

The respondents all agreed that usability testing is more about testing how useful the system is for the particular users. Often users were involved in the testing, but in some cases the customers, the persons paying for the system, took part and not the actual users. In some cases the requirement analyzers were responsible for the usability testing. Some of the respondents also mentioned that they asked external usability professionals to do the usability testing. Their rational was that because knowledgeable external testers were available, training people within the company was not needed.

One form of testing mentioned by several of the interviewees was user acceptance. During user acceptance testing the users check if the system is developed according to what they asked for, check that all the calculations are right and accept the system. The users get test cases and go through those step by step, so the tasks that the users get are not as open as in usability testing. If the users see some usability issues they are asked to inform on that, but there is not a formal process of usability testing at this point. User acceptance is done in the end of the project, but sometimes it is done earlier to show the users what has been developed.

4 Discussion

4.1 Testing Practices in Scrum

Many of the studies on testing practices are analytical comparisons. One recent example is the study described in [12], where the expected number of failures detected by different testing techniques is compared. The analytical studies are really interesting from a theoretical viewpoint, but further investigation is needed before these can be applied in practice because the main focus in practice is to minimize the effort and time required to demonstrate that the software is good enough [4]. The negative consequences of this lack of information is that testing remains one of the most expensive activities of the development process and with delivered products of poor quality and low reliability [5].

Scrum has gained popularity for the last five years and one positive result of our study is that 44% of the respondents say that software testing became easier than in parallel or prior process. Also 44% of the respondents say that more testing is done and only 12% say that less testing is done. The results on the reasons for why some testing techniques are used less than others indicate that lack of budget is not the problem for unit/component, integration system and acceptance testing, so these testing techniques seem to be acknowledged as vital. Lack of training or knowledge was a fairly common reason for why unit/component testing, performance/load testing and security testing was used less than other testing techniques.

The results from the interviews indicate that testing in Scrum is easier because the teams in Scrum are often arranged that way that one of team members has the testing role. The team discusses the ongoing tasks on their daily meetings so the testers know well the functionality that is ready for testing. Some of the respondents in the interviews indicated that all the team members know the progress of the tasks in Scrum, so they realize that some testing tasks are pending and acknowledge that.

4.2 Factors Affecting Testing Practices

Various factors that affect the testing processes are emphasized in current research. These include: the involvement of testing in the development process, communication and interaction between the development and the testing teams, management of the complexity of testing, risk-based testing among other factors [13]. Involvement of testing in the development processes is a complicated issue to study, because the software phases glide in parallel, as noticed in a study by Baskerville, et. al. [14]. Their main results show that developers run their testing or quality assurance in parallel with other development phases, which results in adjusted processes.

In a traditional software development process, testing is mainly conducted at the end of the project time. Testing in agile or Scrum development processes differs from testing in traditional processes, such that each increment of code is tested as soon as it is finished, e.g. the testing becomes dynamic almost from the beginning [10].

The results from the survey show that in 36% of the cases, testing is squeezed into the end off each sprint. This indicates that the opposite is true in 64% of the cases that is each increment of code is tested as soon as it is finished. Only 20% of the

respondents say that testing is not well integrated with coding and ends up one sprint behind. Furthermore 72% of the respondents say that testers became more involved throughout the whole development.

Some of the respondents in the interviews mentioned that the testers working so closely with the teams had also some negative consequences. There aren't separate testing teams in all cases, so all the testing is done by team members that sometimes do not have the distance needed to test the important aspects. They are too much involved in the actual development to be able to test the system neutrally. Testing the increments done as soon as they are finished has also the side affect that formal testing of the whole system emphasizing some quality attribute like usability, is not done. One respondent mention that including testers in the team was positive, but separate testing teams were also needed to cover the more extensive testing approaches.

In this study exploring the difference of acceptance testing and usability testing was particularly emphasized because the goal of both of these testing techniques is to determine if the system fits the user's needs. One of the outcomes of this study is that practitioners view these testing types quite differently. Acceptance testing is commonly accepted and done in all cases, but usability testing is not as widely done. The most common reasons for this are lack of time, lack of training and lack of budget.

5   Conclusion

To summarize the results, the fundamental roles, activities and artifacts suggested in Scrum development process are used to great extent in the companies involved and the sprints are usually 2 – 4 weeks as recommended. Potentially shippable product after each sprint is only accomplished in around 2/3 of the cases though, which is less than expected because this is one of the fundamental issues in the Scrum process.

About half of the respondents said that testing became easier in Scrum than in prior or parallel process used and more testing is done. Only a few said less testing is done. The reason could be that one team member has often the testing role in Scrum and the daily Scrum meetings give good overview of what needs to be tested.

Usability testing and performance testing are practiced in a similar way, but unit, integration, system and acceptance testing are much more frequent. Some examples from the interviews show that practitioners would like to do formal usability testing on extensive parts of the system, but because the iterations in Scrum are short and the changes to the system after each iteration are small, formal usability tests do not fit into the project work. When the focus of testing is on quality attributes like usability, security or performance, the examples in the study show that the testers would prefer to carefully plan their tests. Planning and conducting the tests requires extensive workload from the testers, which is in contrast with the fundamental principles in Scrum, simplicity and speed. The implications for further development of usability testing in Scrum are to find ways of testing the usability on a smaller scale so it can be integrated into the testing activities in each sprint.

6   Acknowledgement

7   References

1. Schwaber, K.: Agile Project Management with Scrum, Microsoft Press, (2004).
2. Wixon, D.: Evaluating Usability Methods: Why the Current Literature Fails the Practitioner. *Interactions* 10, 4 (Jul. 2003), 28-34. (2003).
3. Cockton, G.,Woolrych, A.: Comparing Usability Evaluation Methods: Strategies and Implementation – Final report of COST294-MAUSE Working Group 2. In *Maturation of Usability Evaluation methods: Retrospect and Prospect.* Law, E. L., Scapin, D., Cockton, G., Springett, M. Stary, C. and Winckler M. Eds. IRIT Press. Toulouse, France. (2009)
4. Martin, D., Rooksby, J., Rouncefield, M., Sommerville, I. :'Good' Organisational Reasons for 'Bad' Software Testing: An Ethnographic Study of Testing in a Small Software Company. In Proceedings of International Conference on Software Engineering, pp. 602-611., (2007).
5. Bertolino, A.: The (Im)maturity Level of Software Testing. *SIGSOFT Softw. Eng. Notes* 29, 5 (Sep. 2004), 1-4., (2004).
6. Littlewood, B., Popov, P., Strigini, L. Shryane, N.: Modelling the Effects of Combining Diverse Software Fault Detection Techniques, IEEE Transactions of Software Engineering, 26, 12, (2000).
7. Rayside, d., Milicevic, A., Yessenov, K., Dennis, G., Jackson, D.: Agile Specifications, In Proceeding of the 24th ACM SIGPLAN Conf. Companion on Object Oriented Programming Systems Languages and Applications, ACM, (2009).
8. Dybå, T., Dingsøyr, T.: Empirical Studies of Agile Software Development: A Systematic Review. Information and Software Technology, 50, 9-10, (2008).
9. Abrahamsson, P., Warsta, J., Siponen, M. T., and Ronkainen, J.: New Directions on Agile Methods: a Comparative Analysis. In Proceedings of the 25th International Conference on Software Engineering, IEEE Computer Society, (2003).
10. Crispin, L. and Gregory, J.: Agile Testing: A Practical Guide for Testers and Agile Teams. Addison-Wesley, (2009).
11. Graham, D., Veenedaal, E. V., Evans, I. and Black R.: Foundation of Software Testing: ISTQB Certification. Thomson, (2007).
12. Morasca, S., Serra-Capizzano, S.: On the Analytical Comparison of Testing Techniques. In Proceeding of ACM ISSTA, ACM press, (2004).
13. Taipale, O., Smolander, K.: Improving Software Testing by Observing Practice, In Proceedings of the 2006 ACM/IEEE International Symposium on Empirical Software Engineering, ISESE '06. ACM, New York, (2006).
14. Baskerville, R., Levine, L., Pries-Heje, J., Ramesh, B., Slaughter, S.: How Internet Software Companies Negotiate Quality, Computer, 34, 5, (2001).