LSQ: an approach for learning software development with quality

Leandro Guarino de Vasconcelos
Technological College of Guaratinguetá (FATEC)
National Institute For Space Research (INPE)
le.guarino@gmail.com

Luiz Eduardo Guarino de Vasconcelos
Technological College of Guaratinguetá (FATEC)
Flight Test Research Institute (IPEV)
du.guarino@gmail.com

***Abstract*—Due the quick change of business processes in organizations, software need to adapt quickly to meet new requirements by implementing new business rules. For this, many technologies have been created in the field of software development, to accelerate the production and maintenance of software products. However, learning the software development still carries challenges, especially when there is the concern of developing software products with high quality. In this paper, we propose an approach to learning software development with quality, the LSQ. This approach was motivated by the study of factors that influence learning software development, considering how hypotheses interdisciplinarity, Problem Based Learning, methodologies and technologies of software development. The LSQ was applied in the case study of a graduate course related to Web applications. The results were observed from the feedback of students and they allow us to say that the development of interdisciplinary projects with PBL impact positively in learning software development.**

Keywords- Agile Methodology, Problem-Based Learning, Software Development, Test Driven Development, Interdisciplinarity

I. INTRODUCTION

The desire to perform interdisciplinary practices within the context of universities and contextualize every topic of the menus has become a consensus among teachers and educational researchers. Increasingly, the term interdisciplinarity is present in the guidelines of the universities in the official documents and in the vocabulary of the contemporary university.

However, the development of a truly interdisciplinary project in the universities and faculties still face barriers such as: need to reform education [1] [2], lack of experience of teachers for the subject [3], the difficulty in implementing a real integration rather than just include specialists in every subject [4] and overcomes traditional paradigm of discipline-based theoretical frameworks [5].

Moreover, it is necessary to find a more exciting and interesting for learning aimed at Generation Y, because, according to [6, 7, 8], the main feature of this generation that emerged in a time of great technological and economic prosperity is the use intensive and the allure that grow in relation to technology. According to [6], the youth of this generation "live and breathe innovation, constantly seeking to improve the way things are done." Accordingly, the use of PBL (Problem Based Learning) may be appropriate.

The PBL is based on the premise of cognitive psychology, which says that learning is a process of building new knowledge, rather than just receiving them [9]. The student, working with others, analyzes the problem, formulate learning issues and questions, conducts research and research, create hypotheses and find a solution to the problem [10].

The advantages of PBL include (i) a better preparation for solving real problems, (ii) ease of finding information, (iii) practical application of the knowledge obtained in the theory as well as their retention and, from a point view more subjective, (iv) makes the learning process more exciting and interesting.

There are few studies of PBL in IT and software development reported in the literature. Added to this, the courses related to software development still have other challenges, since the technologies, methodologies and software evolve very quickly [11].

In addition to the technologies used, the software development methodology also impacts on productivity and meeting deadlines while developing software. It is known that traditional methodologies (e.g. waterfall) may lead to the detection of failures late, increase the time and cost of development and maintenance of software, once an application is in the testing stage, it is very difficult to go back and change something that was not well-thought out in the concept stage, no working software is produced until late during the life cycle, high amounts of risk and uncertainty, not suitable for the projects where requirements are at a moderate to high risk of changing [12, 13]. On the other hand, agile methodologies are based on Test-Driven Development (TDD), maintains the technical debt under control, maximize the Return on Investment (ROI) and reduce the risks for customers and companies [14]. According to [15], software applications developed through the agile methodologies have higher success rate and lower risk than traditional waterfall methodology.

In this paper, we present the LSQ which is an approach to the teaching-learning process in software development courses for Web using PBL in interdisciplinary projects. The LSQ was empirically developed and applied like a case study in a course of post-graduation.

978-1-4799-3187-3/14 $31.00 © 2014 IEEE
DOI 10.1109/ITNG.2014.62

The LSQ has been validated with the implementation of a survey to students. The goal was to investigate which factors impact the learning of new technologies to software development, considering as main hypotheses PBL and interdisciplinarity.

This paper is organized as follows: In Theoretical section are described briefly the Scrum methodology, the concept of TDD and the relationship between them; following the LSQ approach is discussed and the scenario in which the survey was conducted is presented; the fourth Section shows the results and discussions on the approach, and finally the conclusions are presented and future work.

II. THEORETICAL

Nowadays, there are dozens of agile software development methodologies (e.g. Scrum [16], XP [17], Crystal [18]), and all these methodologies are based on the Agile Manifesto [19]. The Scrum methodology and their variances (e.g. Scrumban, Scrum with other methodologies) is the agile methodology most used in software development projects using agile methodologies [20].

Scrum is used in projects with uncertain requirements and unpredictable risks resulting from the implementation of new technologies and strategies [12, 21]. This methodology has three key roles: Product Owner (PO), Scrum Master (SM) and team members [12]. Requirements (i.e. user stories) should be identified, prioritized and documented in product backlog [12]. After identifying the user stories, the development schedule can be set through the planning poker technique [12]. The development cycle is iterative and incremental, and based on sprints. It is recommended that each sprint will not take more than 30 days [12].

At the beginning of each sprint, the sprint planning should be made. This is a meeting where everyone involved in the project should participate. At this meeting, the PO (along with team) sets which user stories will be developed and delivered. Usually are chosen user stories that are valuable to the business. During the sprint, there are short meetings of up to 15 minutes (i.e. daily scrum meeting) that are performed daily so those involved can follow the development of the project. At the end of the sprint, the Sprint Review happens, which is a meeting for the delivery of user stories for PO. After this, there is another meeting, the sprint retrospective. In this meeting, the SM and the team can evaluate how the sprint was done and suggest improvements for the next sprint. The monitoring is done through of the burndown chart. Figure 1 shows a diagram representing the stages of Scrum.

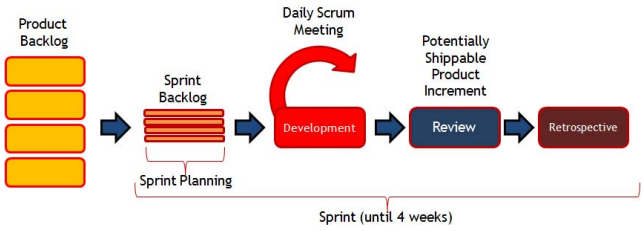

Figure 1 - Scrum methodology. Adapted from [16]

Although the success rate of software delivery is greater with Scrum when compared to the traditional methodologies, it is necessary to use development techniques that allow delivering software higher quality.

Currently, there is a recommended technique for increasing the quality of software that is Test-Driven Development (TDD), which is based on short cycles repeated. Figure 2 shows the workflow of the TDD. First, the developer writing an automated test case that defines a desired improvement or a new functionality. So, it is produced a code that can be evaluated by the test. After this, the code should be refactored under acceptable standards [22]. According to [23], TDD encourages simple designs and inspires confidence code. Through TDD, programmers can apply the concept to improving and debugging legacy code developed from ancient techniques [24].

The design of a testing strategy is, essentially, a process of identifying and prioritizing project risks and deciding what actions to take to mitigate them [25]. Software quality has many dimensions, each requiring a different testing approach. The identification of the testing strategy and, consequently, the tests cases for the PBL was based on the agile testing quadrants [14], presented in Figure 2 and proposed by Brian Marick. The agile testing quadrants are widely used for modeling various test types, which are necessary to ensure high quality of application [22]. In agile methodology, test usually starts in the second quadrant with the Acceptance Test. Such approach reduces the number of faults and improves productivity [26].

Quality is not equal to test. Quality is achieved by appropriately mixing the development and testing activities until one is indistinguishable from the other [27]. This goes back to what was proposed by [28] where the topics of different subjects should be intertwined forming a network facilitator of learning. Joining different content is important but not the only factor of success of interdisciplinary practices. According to [29], the interdisciplinarity stems more from the encounter among individuals than among subjects. This statement is consistent with the first value of the Agile Manifesto [19], which says “Individuals and interactions should be over processes and tools”. Finally, interdisciplinary practice requires appropriate pedagogy, integrating process, institutional change and relationship between disciplinarity and interdisciplinarity [30].

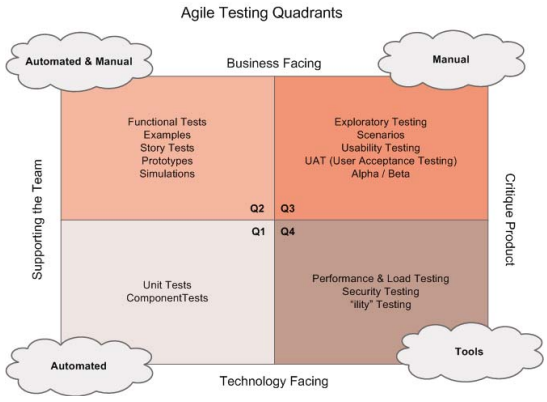

Figure 2 - Agile Testing Quadrants [14]

III. THE LSQ

This section presents the LSQ that is an approach to the use of interdisciplinary and PBL for learning of the software development with quality and describes the scenario in which the LSQ was applied.

A. The Approach

According to [11], technologies evolve quickly. In the software development theme, especially after the beginning of the open source, many technologies have emerged (and appear) for different purposes: accelerate the production of source code, allow easy accommodation of changes, and allow collaborative code management, among other. Although different, these goals are aligned to the continued evolution of business processes of companies, which must be met quickly by management systems.

Given this scenario, software development courses also need to adapt to uniting the fundamental concepts of software engineering with new technologies of development.

However, the available time for accommodate such changes in undergraduate and graduate programs are hard. Therefore, it is necessary to rethink the way software development courses are taught.

We propose an approach, the LSQ, to the software development projects in the undergraduate and postgraduate courses using PBL and interdisciplinarity. The purposes of the LSQ are: (i) align the concepts taught in the classroom with practices of the market, (ii) challenge students to solve problems, (iii) using technologies that support the management team, (iv) accelerate the learning of new technologies. For these purposes to be achieved, we propose the following steps:

1) Plan interdisciplinarity: this phase is essential for that projects developed have satisfactory results. The integration of the subjects must support the entire lifecycle of software development in order to help students overcome obstacles.

2) Stimulating the research for solutions for a real problem: in this step, teachers guide students to pursue a problem existing in companies active in the market or give real problems. This forces the integration of theory and practice. In addition, this phase is crucial for the definition of the software requirements.

3) Choosing a methodology for software development sensitive to changes: the projects are developed in a learning phase, it is common to changing requirements due to the inexperience of students in software development. Thus, the use of methodologies sensitive to changes minimizes the barriers faced during development.

4) Choosing development technologies: in this step should be defined the development platform which consist: programming language, database, Integrated Development Environment, version control software and manage information about the software.

5) Motivating students to implement quality solutions: in this step should be chosen techniques and tools for software testing. This is necessary so that students understand the difference between a functional software and a prototype, and advise them to get quality software.

6) Monitoring the project's development: in this step, teachers can use management tools to help project teams and follow of projects. These tools can also be useful for the continued evaluation of individual or collective.

The next section presents the scenario where the approach was applied in order to validate it.

B. Scenario

This approach was applied in the post graduation course of the "Design and Development of Web Applications", in the second half of 2013, in a private faculty in the state of São Paulo, Brazil. This course has four modules and each module has a set of subjects. All modules are designed so that the topics of the subjects could be intertwined for applied in interdisciplinary projects.

The classes were once a week with a workload of eight hours per day. The Module II had 20 graduates students in fields related to the Information Technology (e.g. Computer Science, Computer Engineering). The 85% of the students were men and 15% were women. The average age of students was 22 years and all worked in the IT field for at least three years.

For this study we used the module II with interdisciplinary project and the application of PBL (Problem Based Learning) in a problem defined by the students. The project included the subjects Agile Methodologies, Web Development II and Software Testing. In the study scenario, 90% of students were unaware of the topics covered in the subjects. The subjects and topics are shown in Table I.

TABLE I. SUBJECTS AND TOPICS

Subject	*Hours*	*Topics*
Agile Methodologies	24	1. Manifesto for Agile Software Development 2. Agile Methodologies (e.g. Scrum) 3. Monitoring Tasks (e.g. Kanban) 4. Version Control Software (e.g. GitHub)
Web Programming II	32	1. Framework Grails 2. Framework ZK
Software Testing	24	1. Test-Driven Development (TDD) 2. Agile Testing Quadrants 3. White-box testing techniques 4. Black-box testing techniques

The following topics detail the implementation of the steps of LSQ.

1) Step 1: Planning interdisciplinarity: In this step, the professors of the subjects listed in Table 1 have defined the scope of the projects, the assessment criteria, the limit of number of students per group and the topics of subjects that would be intertwined in order to support project activities.

2) Step 2: Stimulating the research for solutions for a real problem: According the characteristics of PBL, in this step each team researched, in the first two weeks, a real problem of some company so they could analyze it and develop a Web-based solution. Identified problems and proposed solution were presented at the 3rd week so that professors could validate the scope of the problem / solution, according to the criteria defined in Step 1.

3) Step 3: Choosing a methodology for software development sensitive to changes: The Scrum methodology has been used for the project development. As it is a project developed in the learning stage, use of the Scrum is feasible for applying to projects with unpredictable risks resulting from the implementation of new technologies and strategies [12, 21] and due to the characteristics of PBL, where the requirements are not well known.

Due to the holidays in academic calendar, the module was completed in 12 weeks. This made it possible to split the module in 3 sprints lasting 4 weeks each. We call a Sprint # 0 for learning the theory / practice three fundamental subjects. Following two other sprints were performed (i.e. sprint #1 and sprint #2). In these two sprints, the topics of the subjects continued to be taught. However, part of the class time was reserved for Scrum meetings (e.g. weekly scrum, sprint planning, sprint review and sprint retrospective).

In the week #1 of the module, the students were divided into teams of development with a maximum of 5 members. This division occurred without the influence of teachers and students considered the familiarity and geographical distance among them. As team members were geographically separated, were necessary communication tools such as email, Hangout [31] Google and Skype [32] from Microsoft for virtual meetings.

Due to the personal and professional commitments of students, it was decided that each student will devote 1 hour daily to the project, Monday through Friday. Thus, each student was committed to the project 5 hours per week or 20 hours per sprint.

Being an academic environment, within the team there were changes of Scrum Master. In addition, teachers were considered shareholders in the project so that they could, as well as the PO, have influence the drafting of artifacts (e.g. prioritization of user stories, estimation of user stories) and meetings (e.g. sprint planning, weekly scrum, sprint review). Another adaptation of the methodology was the use of weekly scrum meeting instead of daily scrum meeting, due to the classes occurs weekly.

In the week #4, the teams presented the prioritized product backlog and estimated. The prioritization of user stories was taken by PO of each project. The estimate of each user story was made by the team, SM and teachers of subjects using the Planning Poker technique. In the weeks #5 and #9 were made, respectively, the sprint planning # 1 and # 2. These meetings were held along with the PO and teachers. The PO selected the user stories that would be part of each sprint to determine the sprint backlog of each sprint. In the weeks #8 and #12 were performed sprints review (i.e. presentation of user stories developed for the PO and teachers) and sprint retrospectives.

4) Step 4: Choosing development technologies: In order to analyze the impact of PBL and interdisciplinary in learning of new technologies, the development of the software was done using ZK and Grails frameworks, which are recent technologies and also unknown by students. Grails is an open source, full stack, web application framework for the Java Virtual Machine [33]. It takes advantage of the Groovy programming language and convention over configuration to provide a productive and stream-lined development experience. ZK is an event-driven, component-based framework to enable rich user interfaces for Web applications [34]. In addition to these frameworks, we used a version control software (VCS) that enabled collaborative software development and retention of historical changes in files. For this, GitHub [35] was the tool selected. Due to the incremental-iterative software development methodology, along with the many libraries (e.g. . jar) used during the development, it was necessary the use of a tool to manage the software setup and configuration. For this, the software selected was Maven [36].

5) Step 5: Motivating students to implement quality solutions: In order to motivate students to develop a quality software, we chosen the Test-Driven Development (TDD) technique for software testing throughout the project. TDD forces the verification and validation of the whole unit of code and, consequently, increasing the quality of the shippable. The bugs found during the development cycle of the software were reported in JIRA [37]. At the end of each sprint, the shippable software used the agile testing quadrants. The students performed performance testing and load testing using JMeter [38], acceptance tests using Selenium WebDriver [39] and automated unit tests using JUnit 4 [40].

6) Step 6: Monitoring the projects' development: Each team has also chosen a tracking tool based on Scrum [16] and Kanban [41]. This tool allowed the monitoring of the development of the project by the students, teachers and the PO. For this, the students used the Pivotal Tracker [42].

This is a learning process, thus, assessment is essential to correct the route. Therefore, the evaluation criteria were announced and detailed at 1st meeting. From the week #5, we evaluated the students through continuous monitoring of the participation of each one during the project development. This was done through monitoring tools and Scrum meetings held weekly in the classroom.

IV. MAIN RESULTS AND DISCUSSION

For validating the LSQ, we developed a quantitative questionnaire that was answered by students at the end of the project development (i.e. spring #2), in order to investigate the integration of subjects and which factors impact the learning of new technologies for software development.

The survey questions were drawn from four possible impact factors (hypotheses) on learning new technologies for software development: development methodology; development technology; interdisciplinary and PBL.

The questions were classified into issues Yes / No (YNQ) with answers such as Yes or No; issues Enumeration (EQ) with responses 1-3; impact issues (IQ) with answers that can be "highly disruptive", "harms little "," little helps"," helps a lot "; and issues alternatives (OQ) with selected answers from a list of items. The categories are Methodology (M), Interdisciplinarity (I), PBL (P) and Technology (T).

The questions included in the survey are shown in Table II.

TABLE II. ISSUES SURVEY

Id	*Type*	*Cat.*	*Issue*
1	YNQ	M	Nowadays, would you use an agile methodology in software development?
2	OQ	M	What reasons do you consider the choice of the methodology in issue #1?
3	IQ	M	What is the impact on the use of a new methodology in the project's development?
4	YNQ	I	Do you agree with the application of the interdisciplinary approach to learning new technologies?
5	IQ	I	What is the impact on the use of interdisciplinarity for learning new technologies?
6	EQ	I	Enumerate 1-3 the major difficulties in the development of interdisciplinary projects with 1 being the most difficult and 3 the least difficulty?
7	EQ	I	Enumerate 1-3 most important benefits in the development of interdisciplinary projects, with 1 being most important and 3 the least important?
8	YNQ	P	Do you agree with the use of a company /project real (PBL) to learn new technologies?
9	IQ	P	What is the impact of using a company / project real (PBL) to learn new technologies?
10	EQ	P	Enumerate 1-3 major difficulties in the projects' development with a real company, with 1 being the most difficult and 3 the least difficulty?
11	EQ	P	Enumerate 1-3 most important benefits in the projects' development with a real company, with 1 being the most difficult and 3 the least difficulty?
12	IQ	T	What is the impact on the use of a new programming language in the project development?
13	OQ	T	What is the main advantage in learning a new programming language (e.g. Grails) ?
14	OQ	T	What is the main difficulty in learning a new programming language (e.g. Grails)?
15	IQ	T	What is the impact of the collaborative code on the project?
16	IQ	M	How relevant is the use of tests in software development?

The survey was printed and distributed to students after the week #12, and it was answered anonymously. In order to clear doubts, a professor has overseen the implementation of the survey, by reading each question before being answered by the students through the Elevator Statement technique [43]. This technique emphasizes that the answers are given in a short period of time, because what is more important will be remembered first. In this case, we use up to 90 seconds for the answers to each question were made.

The results were classified according to the impact factors (hypotheses) analyzed.

A. *Methodology*

For question 1, 100% of students reported that they would use agile methodology for software development. The main reasons mentioned (i.e. question 2) for the choice of the methodology were: customer feedback (89%), delivery time of features (79%). With the agile methodology, the goal is a shippable to be delivered as soon as possible, since it is able and acceptable quality. From this, the customer feedback also happens earlier and hence the impact of the change will be less.

In question 3, 58% reported that a new methodology assists in project development (42% reported that helps a lot and 16% that assists a bit). For tests (i.e. question 16), 100% of students reported that the use of testing techniques assists in software development (84% reported that helps a lot).

B. *Interdisciplinarity*

For question 4, 100% of students agree the interdisciplinary approach, but that does not mean they did not find difficulties. The major difficulties (i.e. question 6) identified were: (i) unavailability of team members, (ii) lack of commitment of the team members, and (iii) lack of cooperation from team members. The major benefits (i.e. question 7) indicated were: (i) better preparation for resolving conflicts and problems, (ii) perception that solving a problem requires knowledge of many subjects, and (iii) interdisciplinary practices promote teamwork. Furthermore, most students (95%) felt that interdisciplinary practices assist in learning new technologies (i.e. question 5).

C. *Problem-Based Learning*

For question 8, 95% of students agree the use of PBL for learning new technologies. The major difficulties (i.e. question 10) were indicated: (i) difficulty in finding a company / real project for development, (ii) unavailability of the customer to answer questions and actively participate in Scrum meetings and (iii) lack of cooperation from team members. The major benefits in the use of PBL were also pointed: (i) increased sense of responsibility (ii) let's practice what is discussed in the theory and (iii) best preparation for troubleshooting. Regarding the impact on learning using PBL (i.e. question 11), 95% of students reported that the use of PBL assists in learning new technologies (74% reported that assists a lot and 21% that little helps).

D. *Technologies*

About the impact on the use of new technology in the development project (i.e. question 12), only 21% reported that helps a lot, the remainder reported that harms. That happened because of the need to deliver working software to the customer. Students believe that if we use the language of module I, more features would be delivered to the customer. However, for the teachers, the process is valued more than the amount of functionality delivered in each software project. Also related to technology, students reported that the greatest advantages in learning a new language (i.e. question 13) were: 63% to increase the field of knowledge and 37% for greater productivity. About the difficulties in learning a new language, the main difficulties were listed: 32% had difficulty finding support material, 32% indicated a lack of awareness of the possibilities of language, 26% for the lack of knowledge of command syntax and 10% for the difficulty to setting up the environment. For question 15, 79% of students considered that the practice of collaborative code (i.e. kept in team) assists in the software development.

The results show that students considered valid the topics covered by the subjects, and they did agree to the LSQ approach for interdisciplinary practices and use of PBL.

V. CONCLUSION

Generally, in learning new software technologies, the students often face difficulties in the integration of theory and

practice. Often, examples in the classroom are not interesting because they don't contain challenges that motivate the research for answers.

In order to offer a solution to the obstacles encountered during software projects in the learning stage, we present the LSQ approach, which assists in learning software development with quality. The main contribution of this paper is the division into steps to software development and integration of agile methodologies and software testing techniques.

The LSQ consists of six steps that include learning based on PBL, interdisciplinarity and new technologies. The results obtained through the case study in the course of postgraduate indicate that PBL and interdisciplinarity are factors that positively impact the learning of software development. In contrast, it was observed that students face technical difficulties when newer technologies are used sometimes by the lack of detailed documentation.

Developing software with PBL requires dedication and commitment to overcome the various obstacles presented, both for students and teachers. The interdisciplinary approach allows students who do not know each other, exchange knowledge and different priorities.

In addition to the results discussed in Main Results and Discussion section, we emphasize that some factors are essential to the success of interdisciplinary projects with PBL, such as: analyzing market trends that will be used in the course and that add value to the experience of students in the short and medium term; intertwining the topics of the subjects in order to maximize the benefits; encouraging meetings between teachers so that they can define and monitor the content of others; and having the support of the educational institution for the flexibility of topics of the subjects.

In future work, we intend to apply the approach with different methodologies and development platforms. Furthermore, it is suggested research on the behavior of teachers in interdisciplinary practices and characteristics necessary for teachers who want to develop interdisciplinary projects with PBL.

REFERENCES

[1] R. M. FELDER, “American enineering education: current issues and future directions”, International Journal of Engineering Education, v. 9, n. 4, p. 266-269, 1993.

[2] J. BORDOGNA, “Systemic change for engineering education: integrated trends in the United States”, International Journal of Enginnering Education, v. 9, n. 1, p. 51-55, 1993.

[3] J. RHEM, “Problem-based learning: an introduction”, The National Teaching and Learning Forum, 1998.

[4] M. HUBY, S. CINDERBY, A. OWEN, “Interdisciplinarity in practice: challenges for research and policy”, SECRA working paper, 2005.

[5] K. A. HOLLEY, “Understanding Interdisciplinary Challenges and Opportunities in Higher Education”, ASHE Higher Education Report. Jossey-Bass, 2009.

[6] D. TAPSCOTT, “Geração digital: a crescente e irreversível ascensão da geração Net”, São Paulo: Makron Books, 1999.

[7] N. HOWE, W. STRAUSS, “Generations: the history of America’s future”, 1584 to 2069, New York: Morrow, 1991.

[8] B. TULGAN, “Not everyone gets a trophy: how to manage generation Y”, San Francisco: Jossey-Bass, 2009.

[9] L. R. C. Ribeiro, “Aprendizagem Baseada em Problemas (PBL): uma experiência no ensino superior”, São Carlos: EduFUSCAR, 2008.

[10] H. S. Barrows, R. M. Tamblyn, “Problem Based Learning: An Approach to Medical Education”, New York: Springer Publishing Company, 1980.

[11] Moore’s Law. http://www.mooreslaw.org/. 1970.

[12] T. Stober, U. Hansmann, “Agile Software Development. Best Practices for Large Software Development Projects,” Springer-Verlag Berlin Heidelberg, 2010.

[13] T. Gilb, “Evolutionary delivery versus the waterfall model,” ACM SIGSOFT Software Engineering Notes,vol.10, no.3,pp.49–61,1985.

[14] L. Crispin and J. Gregory, “Agile Testing: A Practical Guide for Testers and Agile Teams,” Addison-Wesley Professional, 2009.

[15] CHAOS. “The CHAOS Manifesto. Agile Succeeds Three Times More Often Than Waterfall”, The Standish Group, 2012.

[16] Scrum Alliance. http://www.scrumalliance.org

[17] Extreme Programming: A Gentle Introduction. http://www.extremeprogramming.org

[18] Crystal Methodologies, http://alistair.cockburn.us/Crystal

[19] Manifesto for Agile Software Development. http://agilemanifesto.org

[20] VersionOne, “Survey: The 7th Annual State of Agile Development Survey”. http://www.versionone.com/pdf/7th-Annual-State-of-Agile-Development-Survey.pdf, 2012

[21] M. JAMES, “Scrum Reference Card”, CollabNet Inc., 2010

[22] W. E. Perry, “Effective Methods for Software Testing”, Wiley Publishing, Inc, 2006

[23] K. Beck. “Test-Driven Development by Example”, Addison Wesley - Vaseem, 2003

[24] M. Feathers. “Working Effectively with Legacy Code”, Prentice Hall, 2004

[25] J. Humble, D. Farley, “Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation”, Addison-Wesley Professional, 2010

[26] M. Gärtner, “ATDD by Example: A Practical Guide to Acceptance Test-Driven Development”, Addison Wesley, 2012

[27] J. A. Whittaker, J. Arbon, J. Carollo, “How Google Tests Software”, Addison-Wesley Professional, 2012

[28] N. J. MACHADO. “Educação: projetos e valores”, 3. ed, São Paulo: Escrituras, 2000, 158p

[29] I. V. A. FAZENDA. “Interdisciplinaridade: história, teoria e pesquisa”. 10 ed. Campinas: Papirus, 2002. 143 p

[30] J. T. KLEIN. “Ensino interdisciplinar: didática e teoria. In: I. C. A. FAZENDA (org.). Didática e interdisciplinaridade”, 6 ed, Campinas: Papirus, 2001, p.109-132

[31] Hangout. http://www.google.com/intl/pt-BR/+/learnmore/hangouts

[32] Skype. http://www.skype.com/

[33] Grails Framework. http://grails.org

[34] ZK Framework. http://www.zkoss.org/doc/devguide/ch01s03.html

[35] GitHub: build software better, together.. https://github.com

[36] Apache Maven Project. http://maven.apache.org/

[37] JIRA. https://www.atlassian.com/software/jira

[38] Apache JMETER. http://jmeter.apache.org/

[39] SeleniumHQ: browser automation. http://docs.seleniumhq.org

[40] JUnit. http://junit.org

[41] D. J. Anderson, D. G. Reinertsen, “Kanban: Successful Evolutionary Change for Your Technology Business”, Blue Hole Press, 278 p, 2010

[42] Pivotal Tracker. http://www.pivotaltracker.com

[43] Sutherland, Viktorov, Blount, "Adaptive Engineering of Large Software Projects with Distributed / Outsourced Teams", in International Conference on Complex Systems, Boston, MA, USA, 2006