On the Fly Design: Evolving a Collaborative System to Support Active Learning

Fabrício Matheus Gonçalves, Julio Cesar dos Reis
Alysson Prado, M. Cecília C. Baranauskas

Institute of Computing, University of Campinas (UNICAMP), Brazil

{fabricio.goncalves, julio.dosreis, aprado, cecilia}@ic.unicamp.br

Abstract. *Technologies in the current networked world have supported new ways of communication especially for the young; learning and teaching in online platforms could take advantage of them. In this article, we argue that if we are to design solutions that make sense and meet the different and complex communication needs of stakeholders in educational scenarios, we need to include them in the design cycle. We propose a method to involve the interested parties during the whole software lifecycle, which we are calling "on the fly design". The method has been experimented in real scenarios of Computer Science disciplines. Results show the effectiveness of the proposal in the building and evolving of a collaborative system to support active learning.*

1. Introduction

Software systems to support educational practices have ranged from instructor-developed models intended to help students learn a specific subject, to full-featured commercial or open-source environments such as extensible LMS (Learning Management Systems). Although ambitious, these systems for enhancing course management and student learning have suffered from actual acceptance and adoption. In a survey conducted by Rößling et al. (2008) to investigate problem areas and obstacles to the adoption of such learning environments, we highlight the most relevant aspects including: (1) flexibility as *they constrain the user to the developer's own idiosyncratic choices of course design*; (2) coverage as *the used tools are very nice for what they do, but together they do not cover the full spectrum of what is required*; and (3) cooperation as *data exchange with other tools is also less than good* (p. 144).

As critical factors in determining whether or not these software tools would succeed, results of the same survey suggest: (1) continuous and two way feedback (*to provide students with feedback and instructors with ways of assessing the performance of their students*); (2) usability (*far too many clicks required to perform common tasks, with significant delays to perform useless computations thrown in for added aggravation*), including difficulty to learn how to use the system; (3) peers collaboration (*team interaction tools that assist both synchronous and asynchronous team work*). The respondents of the referred survey also manifested concern with the openness of the environment regarding its code and free access.

The basic assumption explored in this investigation is that existing software systems to support educational practices, in general, have not been constructed taking the stakeholders - teachers, students, institutions - along the design and development process.

DOI: 10.5753/cbie.sbie.2015.394

As a consequence, they may fail in considering specific requirements and adequate refinements in the software. This might be due to the complex necessities and interpretation of end-users taking part in the learning and teaching process. While iterative software development methodologies and the agile process in particular seek to favor the rapid response to changes in the context, they find difficulties in: (1) integrate (re) design and prototyping development cycles; (2) involve stakeholders more directly in the design and review loop.

This article proposes ways of incrementally evolving a collaborative system in an agile process with the participation of stakeholders along the process. We aim at engaging main end-users of an educational scenario in the evolution of a collaborative system capable of supporting active learning through communicating activities and discussions. In summary, we make the following contributions: (1) we propose a method to evolve collaborative systems taking the view of several stakeholders into account; (2) we develop and conduct a case study to investigate the method and its tools, with the participation of undergraduate students in Computer Science.

In the proposed method, we provide ways of considering cyclic rounds of interaction analysing the stakeholders' points of view. This enables the dynamic detection of new requirements and constraints over time. Based on the agile perspective, we define different types of rounds including tasks of design, development and validation. We apply the method in a real-world case study involving students, assistants and teachers from several instances of a Computer Science discipline. Results of the study revealed the adequacy of the approach to support the direct engagement of the interested parties, which increases the chances of adoption of the emerged solution, potentially improving active learning. Our findings indicate that stakeholders have benefited from early releases while providing important feedback for the system evolution.

2. Background

This research is based on the active learning approach, which describes pedagogical strategies focusing on higher-order thinking while engaging students to develop their own understanding through activities and/or discussions (Freeman et al. 2014). Aiming to achieve this goal, we have relied on a set of theoretical educational references that represent the basis for the definition of our model for teaching and learning. More specifically, we aim to combine the Just-in-Time Teaching (JiTT) and Peer Instruction (PI) methods, applying them in an agile process. JiTT (Novak et al. 1999) is a teaching and learning strategy designed to assess students prior knowledge and use it to promote Active Learning in a formative feedback loop. This strategy uses "WarmUps", short pre-instruction assignments, to prepare students thinking to upcoming lesson and unveils hints on students' needs and interests. PI stands for an interactive teaching method exploring conceptual questions to prompt structured peer discussion around a subject (Mazur and Hilborn 1997; Crouch and Mazur 2001). We understand these pedagogical strategies in the light of Open Source and Agile processes. All teaching and learning artefacts are seen as early releases of working in progress educational resources. Thus, all the subproducts of learning may consist of an independent lifecycle, which are reused and remixed in new instances and contexts and may be shared back into the web like a piece of open source or creative commons artwork.

Our early contributions have shown ways of articulating JiTT and PI in an Ag-

ile way. For this purpose, we defined the Social Meaning-Construction Loop (SMCL) (Gonçalves et al. 2014). In this model, the work routine is characterized by three distinct moments: (1) before class "WarmUps", collaborative exercises on the next class subject; (2) the active learning classes with lectures enriched by group and class discussion based on "WarmUps" results or lightning talks, short student's presentations on results of practical activities; and (3) after class peer-review of the contents with student authored quizzes and asynchronous peer instruction (Figure 1). Each iteration consists in input of discipline subjects and output of achievements and built artefacts. Teachers, students, assistants, and institutions define the main roles played by the stakeholders. The micro-interaction is the heartbeat, where formative feedback and peer instruction takes place.

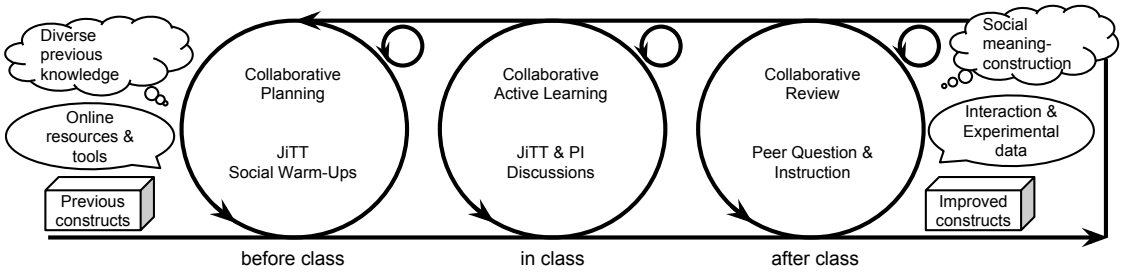

Figure 1. Social Meaning-Construction Loop model instantiated with JiTT and PI.

The proposed model intends to lead PI beyond classroom limits with the help of collaborative systems and to give students the chance to create and improve their own questions. It also aims to give the resulting artifacts an extended lifecycle, with reuse and remix between classes, disciplines and periods. For example, the questions elaborated by students to review content can also be a good warm-up exercise to the next instance of the discipline; the pre-class warm-up project can be revisited after lecture with more formal challenges; and finally a hard Quiz question can be brought back to class to review.

3. On the Fly Design method for system evolution

Inspired by Lean principles applied to Agile methodology of software design, we propose the "On the Fly Design". The Lean principles aim to empower all the interested parties to become problem solvers in continuous learning and continuous improvement of products and process. Within Agile processes, the requirements and solutions evolve through collaboration of self-organizing, cross-functional teams.

We combine and extend these concepts beyond software development to an evolutionary system design, where users are members of the team. They might act according to their capabilities to improve overall system solution. The proposed method involves: 1) stakeholders engagement; 2) a shared view of the system; 3) evolutionary development; 4) listening to the feedback in a situated context; and 5) micro-iterations. These steps iterate towards the desired outcome prioritizing the features that most benefit users according to their valoration (Figure 2).

The stakeholders are at the center of the design and development cycle. They have active voice, leading to innovation in the design process, and review of the development/evaluation stage. In this approach, we emphasize the necessity of sharing vocabulary and perspectives about the addressed problem and the expected solutions. **A shared view**

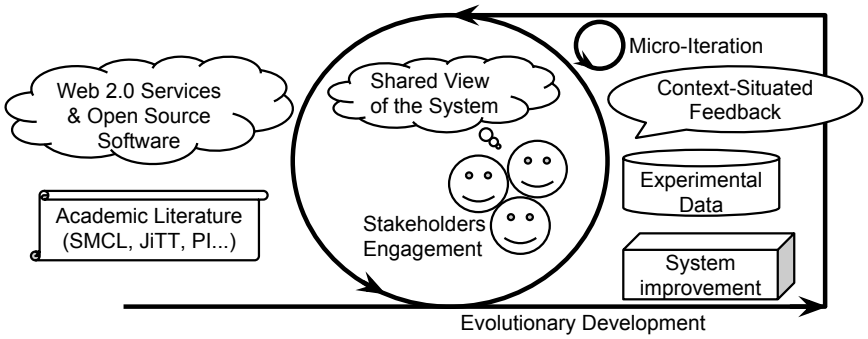

Figure 2. On the Fly Design method.

of the system refers to the guide to where to go from the current software release. It is constructed around the academic state-of-the-art literature and the available materials in the project. This stage provides a clear picture of an anticipated desired solution and acknowledges that the involved trade-offs are the bases to reach stakeholders' commitment on plausible goals to the next iteration of software increment. **The evolutionary development** refers to the frequent releases of the in development software via emergent requirements and architecture. This stage uses the best practices of the Agile development process and the available resources and services of the Web 2.0 and Open Source. **Context-situated feedback** is the main stage where products and processes are refined through reflection and analysis. The reflection is situated in real use scenarios, constructed artefacts and experimental data. Our proposal makes extensive use of **micro-iterations** where all the mentioned steps occur in a very short time frame. Hot fixes or features are added "on the fly" based on implicit or explicit feedbacks that can be spontaneous or stimulated. Ideally, these micro-iteration changes happen when there is less intensive use of the system and they are monitored to evaluate its efficacy.

4. A case study on Wikispaces as basic platform

We conducted an experiment involving three subsequent instances (with 45, 64 and 50 students) of an introductory discipline on Human-Computer Interaction (HCI) at University of Campinas, Brazil. These instances involved undergraduate students in Computer Science and Computer Engineering. They included undergraduates in part-time and full-time with several of them in trainee programs. A graduate discipline on advanced topics on HCI (22 students) has also participated and used the system. Two professors alternated as responsible by the course each semester, and each course instance was supported by different instructors (4). The disciplines had 60 hours divided in 2 classes of 2 hours each.

We applied our method considering a Wiki system as platform. Wiki is a web application which allows collaborative edition of its contents and structure. Wikis are convenient for collaborative knowledge base construction. They have helped reduce technological burden on instructors to apply JiTT methodology (Higdon and Topaz 2009). Wikispaces Classroom[1] is a Wiki platform customized for educational use, free for teachers and students. It supports students groups, subprojects with fine grained access control, calendar and deadlines. This also allows users to integrate several multimedia formats and Web 2.0 applications to enrich text.

[1] www.wikispaces.com

We carried out most of the activities in group at the class or the laboratory, where we performed observations in situated context. Students could formulate their feedbacks on the current release of the system. At any point, users could fill a retrospective form to help improve next activities or add suggestions on how to improve past activities for a new semester. This form was based on agile retrospective meeting questions. It is generic enough to get feedback on learning-support activities, software products and process.

4.1. Results

We present the four main stages of the method, which express the most representative landmarks of the system evolution process in the case study. To support the teaching and learning activities in the first instance of the course, we started developing an improved architecture in the wiki system, to handle the basic requirements of active learning. We evolved the system to accommodate the peer question functionalities adapted for a wiki. In the next stage, further interactive mechanisms were implemented to make easier the interaction for students according to their requirements. Finally, we addressed and improved usability issues according to the results of an heuristic evaluation.

An initial architecture for active learning based on the proposed platform. In the first stage, we needed to support requirements from a Computer Science discipline for a Web collaborative learning environment. To this end, we adopted Wikispaces, which addresses these early requirements as an organizer of course material and the SMCL requirements to support collective authoring and ownership around new course artifacts. The integration of multimedia formats in Wikispaces enables instructors and students to create and recreate meaning into their activities from previous activities or other content.

With the ongoing activities, we realized that the studied context required further an easier way to aggregate and review students created artifacts. For this purpose, we developed and integrated in the original wiki system an architecture aiming to mashup wiki content and online spreadsheets. To implement this, we created a Google Apps Script (GAS) to capture all students contributions into Google SpreadSheets[2], where instructors can review and synthesize this material to be presented in the next class. Therefore, all the Wiki contributions are captured into a spreadsheet through a Really Simple Syndication (RSS) feed reader script. Afterwards, they are reviewed and synthesized into tables and graphics, and then embedded back into Wiki pages to work as formative assessment to students (Figure 3). This sheet was also used to facilitate grading students' contributions.

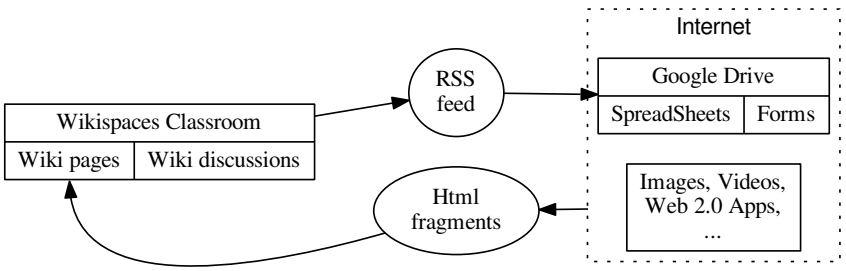

Figure 3. Proposed architecture showing the mashup of Wikispaces, Google Spreadsheets and rich Web 2.0 content.

[2]www.google.com/drive/

The initial architecture accommodated the means to deploy several pre-class activities exploring JiTT, including a social communication among the students discussing on each other's solutions to the tasks before class. The architecture also supported the implementation of further features to guide students' group work regarding in-class assignments with online group review of each other's work before in-class presencial discussion.

Supporting Peer Question and Instruction via the proposed platform. Our proposed model requires after class review of content through a PI variation called *Quizzer* activity. This variation consists in asking students to create multiple choice questions, based on the topics discussed in class, and then review each others questions to find and justify the correct answer. This activity promotes clarifications since students might suggest improvements to each others' questions and answers.

The adaptation of such complex workflow to wiki's authoring and discussion system revealed the need to circumvent some limitations of the initial architecture to create facilities to users. For instance, students needed to visit an external server page before elaborating a new question to obtain information such as: (i) an unique question number; and (ii) a random correct choice.

At this stage the *Quizzer* release was delivered as a detailed multiple steps handout enriched with an evolving table of do's and don'ts based on students' needs and difficulties detected by instructors. Despite the improved support to help students elaborating the questions, it still required manual work including students' auto-organization seeking to cover all questions with revision. Also, it forced authors in revisiting the questions after some peer discussion to inform his/her opinion on the right choice.

Making the system more interactive. As part of the discipline practices, students were taught to perform a Heuristic Evaluation (HE) of usability (Nielsen 1994) to identify issues related to design of user interfaces. To practice this new skill and taking the stakeholders involvement into account, the students were asked to carry on an HE of Wikispaces and the *Quizzer* activity. This was conducted as a group activity at the laboratory, where students should simulate the *Quizzer* activity and report on the main difficulties guided by the heuristics.

This task helped us to examine the system understanding students' main concerns. Among others, we observed the lack of control as they were unable to remove contributions (a Wikispaces design choice) or yet the lack of a convenient way to see his own and his friends' contributions aggregated in portfolios. More directly related to the *Quizzer* experience, users reported as the main limitation, the navigation between different questions elaborated by users. Also at this stage, the real use of the wiki by the students revealed some performance issues that could interfere on the interaction. For example, the Wikispaces velocity degradates when students open too many windows simultaneously, like one per question. Some of these issues were quickly addressed; for example, by providing links to a wiki search by username, which facilitates finding and reviewing each others contributions.

Wiki systems usually do not expose more complex programmatic control and interactivity to the user. In the conducted case study, this brought many issues to the students and they required further interactive mechanisms in the *Quizzer*, mostly related to the navigation between questions. Coupling with that demand, more interactive solutions

on the client side were proposed, such as to enable the view of authors' opinions after responding the question. Two major solution options were possible in this context. First, the wiki as a web page could be scripted on the client side with a browser plugin. Second, it may also be scripted with a javascript plugin hosted in the server with the content and executed in the client browser to provide extra wiki functionalities such as better content interactivity. To address this issue aligned with the students' necessities and options of use, we favored and implemented the second solution, because it requires no extra effort on the user to install a browser plugin. This plugin uses jQuery[3] javascript library to convert a static wiki page into an interactive single page application. It loads dynamic content using ajax requests to fetch randomized questions and answer list from a GAS, loading also some random question text and discussions from the wiki.

Based on the users' needs elicited during the situated class activities, we implemented several other features to improve the interaction. The interactive *Quizzer* changes the color of the clicked answer alternative according to the expected response. In particular, to the red color when it diverges from author choice or yellow otherwise, automatically following to the discussion or the next questions on each case (Figure 4). The yellow color was chosen to indicate that agreeing with the author does not make the answer right since the questions are still in development and need peer review.

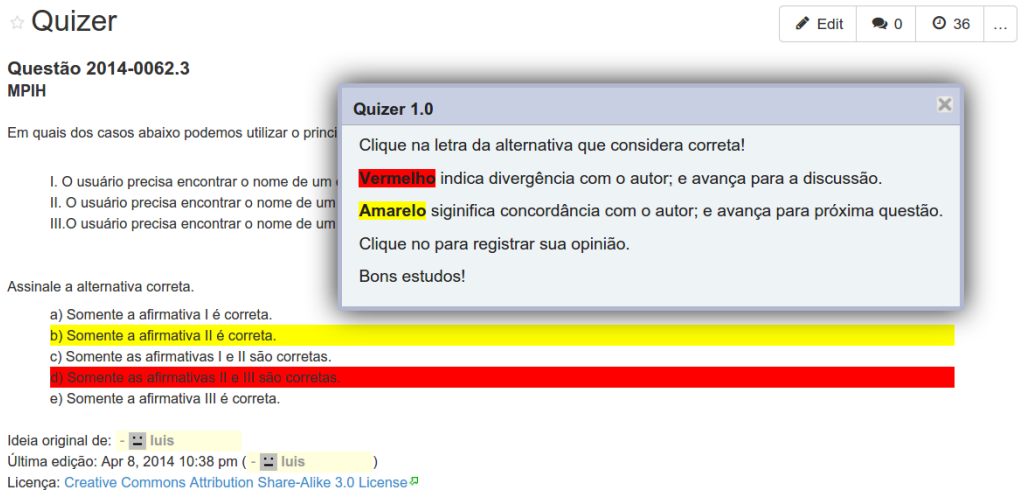

Figure 4. Interactive *Quizzer* single page application with clickable question's alternatives (in original language).

This prototype revealed the necessary features to help students review the questions before examinations. This pointed out a significative improvement over the initial manual approach. Students who performed the HE in this iteration did not report on new complaints about further usability limitations. As the semester ended, this version was brought to the next iteration without major changes. Furthermore, the record of attempted alternatives for each question could be explored as an alternative way to find the hardest or controversial questions and be the basis to explore learning analytics as an extra benefit.

Improving the usability of the proposed platform. In the next semester of the case study, the new class started using the evolved system (the achieved prototype re-

[3]jquery.com

lease). Students were invited to perform a new HE on the interactive *Quizzer* version. Their expectation was higher with respect to the system and several suggestions for improvements were raised as well as limitations and usability violations.

After a fast release addressing the main complains found via HE, students also provided several spontaneous feedback using the system's feedback form. This guided the fine tuning of some key usability aspects. First, we prioritized fixing the HE violations. Second, we transformed textually described handout steps and policies in interface elements interacting and guiding the user through the activity. For instance, we implemented a mechanism to stop the automatic change to the next question. This provided the users with the possibility of interacting with others' answers, even when hitting the "right" answer.

At this stage, we also developed a feature that enabled students to evaluate the question using the same rubric of the instructors with a star vote component (Figure 5; left). Supplementary functionalities addressing the users' evaluation included: (1) automatic switch of focus to discussion corresponding to the selected alternative, and hints about the expected discussion as justification, synthesis or complementation of previous answers (Figure 5; right); (2) a form to elaborate new question that automatically fetches the necessary data and preview of available templates; (3) automatic publishing of elaborated questions without manual intervention. This mechanism allows a balanced selection of questions still not seen or discussed, which increases the chance of revision for all questions. This development resulted in the current system release, used in a third semester of the discipline. This version simplifies the process with several facilities to users in terms of interaction, especially regarding question elaboration and review.

Figure 5. User-friendly *Quizzer* with improved usability (in original language): button alternatives and star peer-assessment (left); guided discussion (right).

5. Discussion and related work

Existing methods for software development do not favor the situated design choices and the rapid response to the dynamic educational context. We proposed a solution suited to take interested parties into account to define and implement evolving software features in an agile process. The proposal was applied in a real-world case study. Our method brings several contributions and benefits as described in what follows.

Stakeholders' participation. The defined approach promoted users participation in the whole software lifecycle: design, development and evaluation. They collaborated to achieve a better software solution, cooperating with each other in his own role. The method enables stakeholders to share a common view and vocabulary; and over time, on each iteration, everyone learns a little more about the constraints acting over the system, and have a new understanding on how to proceed. In our case study, the system releases have been used for three semesters by around 180 students, some full professors and instructors. Even if slightly different approaches were taken to conduct the courses, the evolved system was able to support these stakeholders in taking the most of available materials and discussions. This revealed the adequacy of the method to enable the design of appropriate educational software for supporting active learning.

Software evolution and users' feedbacks. The several types of feedbacks from interested parties, such as explicit feedbacks, voluntary and stimulated ones, contributed to the definition of enriched features, which when implemented in an agile process allows the adequate software evolution according to the situated context. Simultaneously, students evolved their contributions according to their understanding of the system and their needs. The developed tools to support students providing semi-automatic feedbacks in their activities helped teachers defining the grades based on a quantitative and qualitative view of users' activities.

Adequate support for active learning. Our method addressed a technological support for the SMCL model. The conducted case study showed the effectiveness of our proposal to make this model feasible in practice. This enabled the appropriate evolution of the system taking several iterative improvements into account to accommodate the combination of the educational models' elements and the interested parties' needs.

Limitations. Although this study contributes in several dimensions to improve online collaborative learning support, we observe that a good version control and testing scheme is prerequisite to the on the fly approach. Further investigations include the level of system and method tailoring by the end users, and the level of adequacy of the system to other learning approaches.

Related work. Despite their value to improve online system learning, the methods and tools that served as a basis for our work, taken alone are not suited to involve the interested parties at the right time in a situated context. Our experimental results have shown effective to address this open research challenge. The involvement of stakeholders "on the fly", along the system design and development, brought essential differences to the product of design and to the underlying learning process. The questions created by the students are multiple choice with five alternatives; unlike the Concept Tests, created by teachers in the PI, the questions can address several concepts at once, and the alternatives are not necessarily focused on the most frequently asked questions, thus requiring more time for appropriate resolution. Moreover, the *Quizzer* embeds a simple peer assessment and learning analytics that identifies the most controversial/interesting questions for in-class discussion, leaving the simpler to individual work. Going beyond the collaboration modelling proposed by Rangel et al. (2011) and the web 2.0 orchestration by Lin and Jou (2013), our approach systematically involves stakeholders while integrating (re) design and prototyping development cycle in a methodological approach.

6. Conclusion

Contemporary educational systems require adequate and novel software development processes to meet the dynamic and complex requirements of stakeholders; such processes and their practices have not been extensively explored in literature. In this article, we proposed a method named "On the fly Design", inspired by the agile development process, to design a collaborative system in support of active learning. A case study in evolving a collaborative software in a real teaching and learning situation was carried out to explore the proposed method. This case study illustrates the method in use involving students in several instances of a Computer Science undergraduate discipline. The obtained results suggest the effectiveness of the proposed method to engage and benefit the stakeholders considering the adequate evolution of the software functionalities via relevant feedbacks. As future work, we plan to investigate how to empower students with learning analytics techniques. so they can thus self-regulate their learning efforts with information on previous performance, and compared with peers.

Acknowledgments: We thank National Council for Scientific and Technological Development (CNPq) (grant #136239/2013-7 and #308618/2014-9) and São Paulo Research Foundation (FAPESP) (grant #2014/14890-0) for funding. We also thank the anonymous referees for constructive comments.

References

Crouch, C. H. & Mazur, E. (2001). Peer instruction: ten years of experience and results. *American Journal of Physics*, *69*, 970.

Freeman, S., Eddy, S. L., McDonough, M., Smith, M. K., Okoroafor, N., Jordt, H., & Wenderoth, M. P. (2014). Active learning increases student performance in science, engineering, and mathematics. *Proceedings of the National Academy of Sciences*, *111*(23), 8410–8415.

Gonçalves, F. M., Arpetti, A., & Baranauskas, M. C. C. (2014). Facilitando a construção social de significado em sistemas de aprendizado colaborativo. In *Proceedings of the xix international workshop on educational software, tise* (Vol. 10, pp. 318–326).

Higdon, J. & Topaz, C. (2009). Blogs and wikis as instructional tools: a social software adaptation of just-in-time teaching. *College Teaching*, *57*(2), 105–110.

Lin, Y.-T. & Jou, M. (2013). Integrating popular web applications in classroom learning environments and its effects on teaching, student learning motivation and performance. *Turkish Online Journal of Educational Technology*, *12*(2), 157–165.

Mazur, E. & Hilborn, R. C. (1997). Peer instruction: a user's manual. *Physics Today*, *50*(4), 68–69.

Nielsen, J. (1994). Heuristic evaluation. *Usability inspection methods*, *17*(1), 25–62.

Novak, G. M., Patterson, E. T., Gavrin, A. D., Christian, W., & Forinash, K. (1999). Just in time teaching. *American Journal of Physics*, *67*, 937.

Rangel, V. G., Cury, D., & de Menezes, C. S. (2011). Vcom: uma abordagem para modelagem de ambientes colaborativos para apoiar a aprendizagem. In *Anais do simpósio brasileiro de informática na educação* (Vol. 22, *1*).

Rößling, G., Joy, M., Moreno, A., Radenski, A., Malmi, L., Kerren, A., ... Korhonen, A. et al. (2008). Enhancing learning management systems to better support computer science education. *ACM SIGCSE Bulletin*, *40*(4), 142–166.