2006-1770: EXPERIENCES WITH AGILE TEACHING IN PROJECT-BASED COURSES

Valentin Razmov, University of Washington

Valentin Razmov spends time in the classroom as often as he can. He is interested in methods to assess and improve the effectiveness of teaching and learning. Valentin is a Ph.D. candidate in Computer Science and Engineering at the University of Washington (Seattle), where he received his Masters degree in Computer Science in 2001. Prior to that, in 1998, he obtained a Bachelors degree with honors in Computer Science from Sofia University (Bulgaria).

Richard Anderson, University of Washington

Richard Anderson is a Professor in the Department of Computer Science and Engineering at the University of Washington. He is currently the associate chair for educational programs. His main research interests are in Educational Technology and Computer Science Education.

Experiences with Agile Teaching in Project-Based Courses

Abstract

In this paper we describe an agile teaching methodology as applied to project-based software engineering courses. We take the term "agile" from the popular software development methodology that emphasizes short feedback cycles, flexibility, and direct involvement of the customer. Our software engineering courses are goal-driven, we include structural mechanisms to support feedback, and we design the projects around frequent checkpoints. The course content is adaptable to student needs and changing situations. After giving an overview of agile teaching, we address specific issues of course structure for supporting feedback, how we act on student feedback, and the mechanisms for collecting just-in-time feedback. The paper concludes with a discussion of the impact on instructors and students, as well as of results from polling our colleagues on their teaching practices in similar courses.

1. Introduction

Feedback is important for adaptation and learning. Instructors who receive feedback can more effectively tailor their teaching to student needs. Students who receive feedback have an opportunity to see more ways to improve, because there are more open (feedback) channels to offer them guidance. Therefore, more *frequent feedback* can translate into more opportunities for both sides.

Learning is a *continuous process*, the individual steps of which may often be imperceptibly small to the learner[7]. Its intensity depends on at least two factors: the student's engagement ("doing") and the amount of feedback – positive affirmation or corrective guidance – that the student receives. To provide tailored instruction and relevant advice, instructors in turn need to be aware of student needs and how they evolve over time, so instructors themselves have to *seek feedback* in order to stay current.

The premise of our work is that increased student involvement and relevance of classroom discussions and projects leads to improved learning. As students hit roadblocks, they discover areas where they lack knowledge and skills. If instructors have a mechanism to find out that this is happening, they can take advantage of such teaching moments – when *goal-oriented*, *targeted feedback* can be particularly effective, since it would address an existing need.

Typically, however, not all students hit the same roadblocks, and certainly not all at the same time. Each individual comes with a unique background and learning style, so there are clear benefits to a *personalized approach to teaching*. Furthermore, with different challenges faced by different students, a *flexible approach* that takes into account those differences in the learners' needs would produce superior results. As Kent Beck put it[3], "noticing when a learner doesn't have a tool they need or isn't using a tool they already have" is key to effective teaching.

Agile teaching is the term we use to refer to teaching approaches that exhibit all of the above characteristics: emphasis on the continuity of the learning process, goal orientation, seeking feedback from students, flexibility in responding to student needs, a short feedback cycle, and demand-based personalization of what is being taught. The phrase "agile teaching" is derived from the name of a modern software development methodology[5], characterized by short feedback cycles, frequent involvement of customers in making decisions, and flexibility to quickly adapt in response to changes in customer needs. (In the education scenario, customers are our metaphor for students.)

Agile teaching is what teaching assistants often do in seminars and office hours, where the environment makes it easier to have closer interaction with individual students. It is aimed at eliciting and addressing specific questions that students have, thus complementing the more rigid delivery of material in a conventional classroom environment. However, not all courses are characterized by inflexibility of structure in the larger classroom. In many project-based courses, including capstone courses, the emphasis is away from content coverage in lectures, and instead falls heavily on student learning experiences in the process of working on projects. This in turn marks a shift in what instructors can spend class time on, allowing them substantially more freedom to follow the energy of the class – adjusting to student needs and providing advice on the pressing problems students are facing.

In this paper, we focus on project-based courses, examining the role that agile teaching can play in helping students to acquire and retain knowledge and skills. Our own experiences come from project-based courses in software engineering, as well as capstone courses in tablet computing and software development. We also share our findings from surveying colleagues who have taught a range of project-based and capstone courses.

2. Agile Teaching Practices

As was already mentioned here and in the literature[4], the agile teaching philosophy rests on actively soliciting student feedback and promptly reacting to it in order to increase the relevance of topics and advice directed toward the particular student (or group of students), with the ultimate goal of positively affecting student learning. To sustain this process, an instructor must actively and frequently solicit student feedback in a variety of forms.

Deferring the discussion of specific feedback mechanisms to the following section, here we briefly describe the core agile teaching practices we have adopted in our courses. These practices fall into two broad categories, relating to the key questions of how to structure the course and how to react in response to feedback.

2.1. Structuring the Course

In preparing a course, our goal is to embed sufficient structural support into the curriculum to set the stage for regularly obtaining and providing feedback. Specifically, we design our courses around experiential learning, where students work in teams of 5-7 on term-long projects, and –

following the incremental delivery approach[8] with a short iteration cycle – we set intermediate project milestones every 2-2.5 weeks.

Within each iteration there are several scheduled events that relate to obtaining and providing feedback:

- *Post-milestone project discussion meetings*. Shortly after each milestone, the instructors meet with each team for 30 minutes to clarify any issues, discuss their impressions on the state of the project and make concrete suggestions, as well as to gauge student spirits and inquire about problems or concerns. This interactive meeting replaces the more traditional practice of instructors evaluating the merits of student projects while having limited visibility of the context, followed by writing a "verdict" email to the team about the instructors' opinion on the state of the project.

  The informal atmosphere of these project meetings is critical for putting students at ease and for achieving rapport with them – letting them know that this is not about grades, but rather is an opportunity for them to learn – as well as for allowing students to honestly share their ideas and problems without fear or discomfort. We have also found that those project meetings are a very time-efficient way for instructors to form a truthful picture of the state of projects and project teams; in the same amount of time instructors would be unable to reach that depth of context if it were not for the discussions with the project teams.

- *Post-milestone in-class retrospectives*. When the class size is small enough or when there are few project teams, in the class session immediately following a milestone delivery (while impressions were still fresh in everyone's minds), the instructors conduct project retrospectives by asking students about aspects of their work or process during the latest project iteration that they would like to sustain (i.e., continue doing well in the future) and aspects that they would like to improve on. Actual artifacts from two of these discussions are shown in Figure 1.

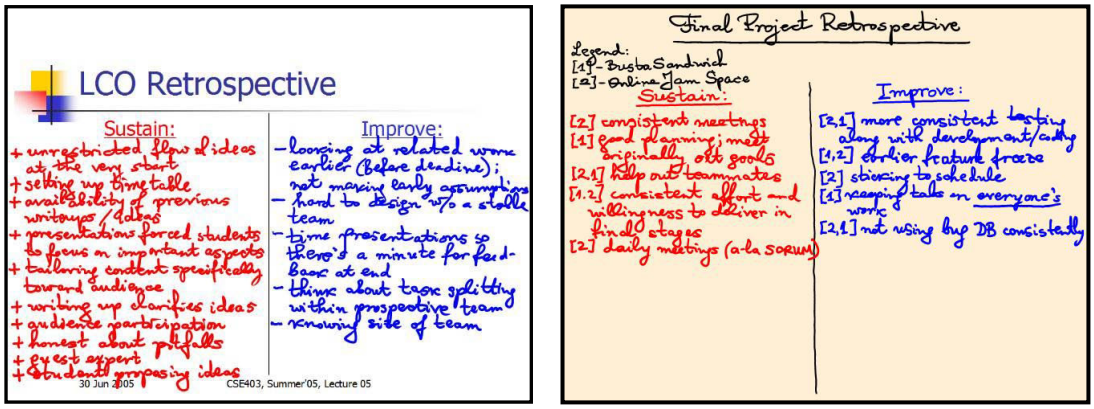

Figure 1. Digital ink artifacts from in-class retrospectives that followed the Sustain / Improve technique. The notes were taken by the instructor, while the students were reflecting aloud. The technology behind creating these artifacts is a Tablet PC application that integrates ink with slides.

Variants of this practice are described in the literature[6], where it is suggested that from a psychological standpoint it is important to address the positive aspects (i.e., the 'Sustain' column) first, before moving on to discuss what needs to be improved (i.e., the 'Improve' column).

- *Post-milestone anonymous peer reviews*. The week after each project milestone, and preferably shortly after a project retrospective, we ask all students in the class to complete an online peer review questionnaire, providing feedback to each project teammate and optionally to other students in the class. The feedback format consists of two components for each recipient: a rating (on a scale of 1-5, where 5 is the top rating) and brief free-form constructive comments. The peer review results are available only to the intended recipient and course staff, and the review providers remain anonymous.

  In a second phase of the peer review activity, each recipient rates the usefulness of each peer review they have received. The original review writers are then able to see the extent to which their comments were useful to the recipients.

  We let students know ahead of time that the results of a peer review (which instructors have access to, since they administer the questionnaire) are strictly for their personal improvement and are not used for grading purposes. Instructors participate in the peer review as well.

  The benefits of doing peer reviews are many – students are able to see themselves through the eyes of their peers and have the opportunity to act on any perceived strengths or weaknesses; they also get practice in providing *useful* feedback to others. Doing a peer review multiple times during a term (in our case, after each project iteration) gives students multiple data points, allowing them to track the extent to which their actions affect their peers' perception of them.

For each of the above three types of practices to be effective and genuine, instructors must explicitly "switch hats" – temporarily stepping down from their powerful role as instructors (and evaluators) and taking on the role of facilitators – to avoid being perceived as someone with whom internal project- and team-related information could not be freely shared. Ultimately, there is an element of trust underlying these activities, trust which instructors earn.

2.2. Acting on Student Feedback

In response to student feedback, the instructors try to react promptly and visibly. Whether the particular student feedback relates to course curriculum issues, the coverage of technical content, or the state and needs of a given team project, in preparation for a class session we consider if it is appropriate to adjust the order or the content of what is covered in order to increase the learning benefit for students. The specific teaching practices we use to support such flexibility are to:

- have a pool of candidate topics to cover and as the next class session nears, draw the ones with the highest current priority based on a complex set of constraints – natural ordering of course content, project needs, student questions, time required for coverage, importance to

student learning, etc. Satisfying all constraints is, expectedly, not always possible, and occasionally there are students who are not satisfied with our decisions. In such cases, if we are aware that this has happened or expect that it may happen, we proactively approach such students and explain the reasons for our choice. The goal of doing this is to keep the communication link to all students open even if we may disagree on some aspects.

- decide which sessions may be most appropriate for collecting and giving feedback, including through the use of technology to facilitate this process. Some sessions heavily involve project work – storyboard designs, user interface specifications, etc. – and are not good candidates for incorporating explicit feedback gathering activities. Other sessions revolve around discussing issues that admit multiple valid viewpoints, reinforcing the value of diversity of opinions. This makes such sessions especially attractive for embedding feedback-related activities[9].

3. Mechanisms for Just-in-Time Feedback

We now discuss our mechanisms for collecting feedback from students. There are both formal and informal ways of collecting feedback; the formal ones involve a planned collection mechanism, while the informal ones rely on picking up information from interactions or information sources associated with the class. We will just address the formal ones here, while recognizing that informal feedback is also valuable.

Ideally, feedback mechanisms should be accurate, representative, accessible, efficient, and non-distracting. For accuracy, we want the feedback to convey the true issues and emotions of the students, whether positive or negative. For certain types of information, it would be more likely to see a genuine reflection of student opinions if anonymity is preserved. It is also important that the feedback be representative, showing what the entire class feels, as opposed to giving the floor to just a few vocal students. Some information (e.g., if there are logistical problems with course projects) will affect all students, even if just one or two raise the flag, while the effect from other issues (e.g., opinions on what topics are important) might vary greatly. This means that the feedback mechanisms should also be easily accessible to students in order to encourage participation. Efficiency is yet another consideration, since there is an overhead associated with collecting or evaluating feedback. Finally, the feedback process should be integrated with student projects, rather than distract students from their project work. A more in-depth discussion of these and other desirable qualities of feedback can be found in the literature[10]; some of it applies mostly to the case of providing feedback to students, while other parts concern the scenario of obtaining feedback from students.

Through our course offerings, we have used many different mechanisms for collecting student feedback, not all of which exhibit all of the qualities above. Here we mention several of these mechanisms (some of them are in both directions – for gathering as well as for giving feedback), and then discuss one particular technological approach that helps in both collecting and providing feedback.

- *Minute paper*. The minute paper[2] is a popular classroom assessment technique, whereby a short question is posed (e.g., "What was the most important topic covered in lecture today?")

and students are asked to respond quickly and very briefly. A common implementation of this technique is to pass out index cards in class at the end of a lecture, and collect them after the activity, so that the instructors can review the information before the next class session.

- *Mid-quarter and end-of-quarter questionnaires*. We often use questionnaires to collect feedback at appropriate times during and after a course. This can be done either on paper, or as a web-based survey. Both numerical and open-ended questions are used. A challenge in designing a questionnaire for feedback is deciding what to ask, so that the returned information can be acted upon, as opposed to constituting general quantitative data on how much students liked various aspects of the course.

- *Reflective writings*. These are short individual writing assignments (usually limited to one page in length), where students are asked to reflect on a given aspect of the course or their project. The intent is to give feedback to both student and instructor. The student, in reflecting upon experiences, realizes what he or she has learned and ideally discovers fruitful directions to further improve. The instructor, in reading student writings, gains an understanding of what difficulties each individual student is facing and so can personalize the instruction. As a way to maximize the potential of this technique, the instructor provides written comments on the writing so that the student can see an alternative viewpoint; the student is also asked to respond to those comments, generating an ongoing virtual conversation while ensuring that the student reads and understands the instructor's feedback. More on the motivation and application of this technique has been covered elsewhere[11]. Finally, we note that such writing assignments can become unpopular among students in technical disciplines, if they are too frequent or not properly motivated. In our experience, two or three such assignments in a 10-week course suffice to keep the feedback channels open without raising concerns among students.

- *Anonymous feedback form*s. We often make an anonymous feedback form available for the students through our course websites. Students can use it to provide feedback about both content and operational issues of the course. Usually, only a small number of the students will use the feedback form, so it is not a representative source. Also, although the type of feedback is unconstrained, the social dynamics associated with anonymity tend to sway uses of this mechanism primarily toward complaints – students feel they have little to fear or hide from if they have positive opinions to share, whereas to offer criticism some would prefer to use an anonymous shroud.

One novel approach that we have tried for gathering and giving feedback is through the use of a *Classroom Interaction System*. A classroom interaction system allows networked, digital communication in a classroom; the basic setting is that students have devices that can communicate with an instructor's device. The instructor receives content from students, reviews it privately, and selectively shows it to the class on a public display. (We have used Classroom Presenter[I], a Tablet PC-based classroom interaction system, although for this particular application, having pen-based input was not a critical feature.) The protocol for collecting feedback with a Classroom Interaction System is that the instructor poses a short-answer question to the class such as "Is software different?" or "What one or two ideas discussed today captured your attention and thinking the most?", and gives students a short period of time to

work on their answers before requesting that responses (that students write on their devices) be sent in. In relatively small classes (of up to 30 students) we have used the system to show many of the student submissions to the class, and make relevant short comments.

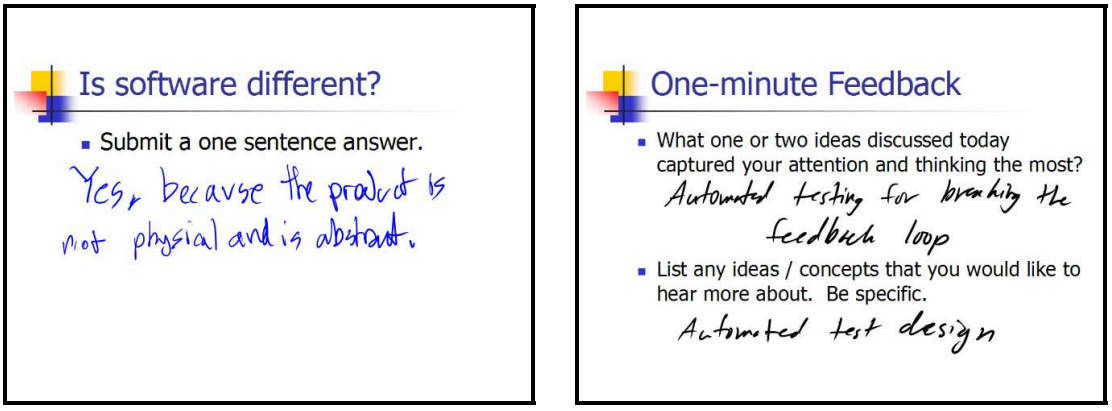

Figure 2. Examples of student feedback collected via Classroom Presenter. In both cases, the questions relate to evaluating lecture content. These activities also illustrate that we were looking for short responses.

There are a number of justifications for our approach of displaying a majority of the responses. This sends a very strong signal to the students that their feedback is valued, and it also gives the audience a shared perspective on the variety of issues related to the question. The public display of the feedback is also helpful in bringing closure to the discussion. Figure 2 shows examples of student feedback received in a software engineering course.

The technological support is important for three reasons. First, the networked communication enables very efficient distribution of activities and subsequent collection of student work, reducing the overhead of the process. Second, the Classroom Interaction System enables the integration of activities into a slide-based lecture, so the design of the feedback questions becomes part of the lecture preparation. Third, and perhaps most significant, the technology allows the written student responses to be displayed as a public artifact, giving the feedback prominence, while also preserving the author's anonymity.

4. Impact

We now turn to the question of what impact agile teaching has on both instructors and students. Data in support of the claims in this section has been gathered through instructor interviews during extensive post-course retrospectives and analyses, as well as from informal conversations with students and detailed course-specific end-of-quarter questionnaires we designed that many students in our courses filled out.

4.1. Impact on Instructors

In this subsection we consider the impact on the authors themselves, when they acted in the role of instructors of project-based courses. A summary of the opinions of other instructors, who have taught similar project-based courses but who do not claim to adhere to agile teaching, can be found in the next section.

4.1.1. Demands on Time

Agile teaching is about working closely with students – to understand their specific needs, to respond to them, and to adjust accordingly when those needs change. There is less guesswork about what students may be thinking – uncertainty is resolved by soliciting and obtaining feedback in regular short cycles. Although the workload on instructors may be higher in comparison to the situation when very little or no feedback from students is requested, this is not a fair comparison since in that case we believe that students would not advance in their learning nearly as much. Agile teaching reorganizes the instructor's time to focus on areas where it matters the most. This does not imply that it is more time consuming or less; indeed, it is the authors' impression that in reality the workload would still fall within the normal expectations.

4.1.2. Satisfaction

Instructors were very satisfied with the interaction aspects of agile teaching and found many of the practices – post-milestone project meetings, in-class retrospectives, student reflective writings, etc. – to be stimulating. One author noted, "I get the satisfaction from seeing how [agile teaching] influences the result – contributing to the [student] accomplishment."

4.1.3. Need for Flexibility

One of the core principles of agile teaching is that of flexibility on the part of instructors. This need may not fit every instructor's teaching style and comfort level. The authors themselves, however, find no problem with it; it matches the demands of the real world, emphasizing an aspect – flexibility – that they try to teach their students to appreciate.

For instructors who teach a course for the first time, the preparation time with agile teaching should not be more than without it. However, some may feel uncomfortable shuffling material without having covered it in class at least once before in the "right order." For instructors who have taught a given course before, the agile teaching approach may add to their preparation time, since reordering content and adjusting the message of individual lecture units means that less material can be readily reused. Still, changes in ordering or spin are rarely fundamental, because the main structure of the course (assignments, exams, project milestones, etc.) is typically in place well before students walk into the classroom and before the first feedback starts arriving.

4.1.4. A Different Mindset

Agile teaching is goal-oriented in contrast with more conventional approaches to teaching that follow a well established content coverage plan. This takes a different mindset from instructors, as well as some time to get used to.

Overall, we feel that agile teaching is a promising approach for project-based courses where the emphasis is on student learning through practice rather than through traditional-style lectures. We acknowledge, however, the concerns that some instructors may have about it.

4.2. Impact on Students

At the end of most courses we conducted extensive questionnaires, aiming to collect data – in both numerical answers and free-form comments – about the extent to which students found value in a set of aspects of the course they took. The questionnaires were not always anonymous (some were conducted online and we were the administrators who set up the data collection and who did post-aggregation of results), but in the cases when we had access to the identities of feedback authors, we had promised the students not to look at the data before the final course grades were turned in. Given that there always was a mix of opinions – including criticisms we had not encountered during the course – we infer that students did not feel overly pressured to say things we may like. On a scale of –2 (strongly negative value) to +2 (strongly positive value), students in the most recent courses rated the agile teaching practices and mechanisms we discussed earlier as shown on Table 1 (shown are average values for each course offering).

Table 1. Class-averaged student perceptions of the value of different agile teaching practices and mechanisms. The scale was from –2 (strongly negative value) to +2 (strongly positive value). All numbers are taken from evaluations of the same course – software engineering – though the details of the individual course offerings differed between terms, including who taught the course; still, in all cases, at least one of the authors was involved.

	Winter 2005	Spring 2005	Summer 2005
Learning through experience	1.31	1.27	1.88
Incremental delivery approach	1.46	1.45	1.62
Post-milestone project discussion meetings	1.23	1.43	1.75
Post-milestone in-class retrospectives	n/a	n/a	1.00
Post-milestone anonymous peer reviews	1.15	1.31	1.38
Iterative format of reflective writing assignments	0.63	1.13	1.38

Note that the averages were not uniformly as high among all questions we asked the students, but these central aspects of agile teaching were generally rated much higher.

Additional free-form feedback on the reflective writing assignments came from a student in an earlier course who found the virtual "conversation" (of responding to the instructor's responses) stimulating and wrote, "The part [of this class] I looked forward to the most besides coding was getting my reflective essay back. [The feedback I got] inspired new thinking."

There are other important evaluation questions on which we do not have conclusive data yet, so we may only offer our own perspectives. As one example, there were frequent remarks in earlier courses about what students perceived to be a disconnect between textbook readings, classroom discussions, and project experiences. These gradually died out and in the last course offering there were none. What we had done to address the problem was – in the spirit of agile teaching – to make it a point to assign specific readings from the book that relate to current project needs (rather than just covering "the next chapter"), as well as to gear classroom discussion toward techniques that can help students make progress on their projects.

Finally, we feel that unlike other novel teaching approaches, agile teaching does not often make students uncomfortable. While we do not have direct answers to the questions "How much do students want to direct how the course goes?" and "How much do students want other students to

direct how the course goes?" – it is possible that some students may feel like the instructor knows best how to direct a given course – the highly positive view they hold of the approach to learning through experience (see Table 1) suggests to us that the students in our courses are willing to take the opportunity to lead and be responsible for their own decisions.

5. Use of Agile Teaching Practices

To get a sense of whether and how other instructors use some of the techniques described in this paper, we conducted a poll of our colleagues who also teach project-based courses. The goal was to understand their attitudes toward aspects of agile teaching. The survey was conducted by email. It was sent to 11 individuals who have taught project-based courses in the past 3 years; we received 6 responses. The results did not surprise us: they naturally showed a range of approaches, with different philosophies, but a common theme of aiming for successful student projects. The faculty generally did not use active methods for collecting feedback from students, but instead relied on sensing student reaction from comments in class and conversations. When feedback was used, it was typically used to adjust subsequent lectures, as opposed to incorporating it into the current lecture. The common reason for using feedback was to fill in gaps in the students' knowledge and to readdress topics which students had not understood. There was a wide spectrum in views on whether the students or the instructor should drive the curriculum, with the extreme points being expressed in the following two quotes: "I've got a collection of topics that need to be covered. How can I possibly rely on the students to direct me toward these topics??? I'll take student feedback into account whenever I detect that I haven't communicated the topic clearly enough." and "After [the first two weeks] the material is pretty much based on questions the students ask or issues I see them hitting. What I like to do is keep a running list of topics to discuss on the lab whiteboard that the students can see and that we adjust as the quarter progresses."

6. Conclusions

In this paper, we have presented an approach to teaching project-based courses that we refer to as agile teaching. The term takes its name from a software development methodology which stresses short iteration cycles and frequent feedback from the customer. In the context of teaching a project-based course, this means that instructors gather feedback from students through various active methods during the course, are flexible in coverage of the course material, and design projects with frequent checkpoints. We have been successful in using this methodology in our classes, and have received positive responses from students about the components of the approach. We consider the main contributions of this paper to be the definition of a methodology for teaching project-based courses and an analysis of various mechanisms to support this methodology.

References

[1] Richard Anderson, Ruth Anderson, Beth Simon, Steve Wolfman, Tammy VanDeGrift, and Ken Yasuhara. "Experiences with a Tablet PC Based Lecture Presentation System in Computer Science Courses." In SIGCSE 2004.

[2] Thomas Angelo, K.P. Cross. *Classroom Assessment Techniques: A Handbook for College Teachers*. Jossey-Bass Publishers, 1993.
[3] Kent Beck. http://c2.com/cgi/wiki?LearningPatterns. (page accessed on Jan 18, 2006)
[4] Andy Chun. "The Agile Teaching/Learning Methodology and Its e-Learning Platform." In Lecture Notes in Computer Science – Advances in Web-Based Learning, Vol. 3143/2004, Springer-Verlag, Heidelberg, pp.11-18.
[5] Alistair Cockburn. *Agile Software Development*. Addison-Wesley, 2001.
[6] Norman L. Kerth. *Project Retrospectives: A Handbook for Team Reviews*. Dorset House Publishing, 2001.
[7] George Leonard. *Mastery – The Keys to Success and Long-Term Fulfillment*. Plume, 1992.
[8] Steve McConnell. *Rapid Development*. Microsoft Press, 1996.
[9] Valentin Razmov, Richard J. Anderson. "Pedagogical Techniques Supported by the Use of Student Devices in Teaching Software Engineering." In SIGCSE 2006.
[10] Valentin Razmov, Stani Vlasseva. "Feedback Techniques for Project-Based Courses." In ASEE 2004.
[11] David Socha, Valentin Razmov, Elizabeth Davis. "Teaching Reflective Skills in an Engineering Course." In ASEE 2003.