

A Course Design and Implementation Experience on Agile Software Development Methodologies

Hongxing Lu
Information Engineering Institute
Zhengzhou University
75 University Road
Zhengzhou, Henan
P.R. China 450052
luhongxing@zzu.edu.cn

Xiaohong (Sophie) Wang
Department of Math. & Computer Science
Salisbury University
1101 Camden Avenue,
Salisbury, MD 21801 USA
(410) 677 5380
xswang@salisbury.edu

Abstract - In this paper, a course design and implementation experience on agile software development methodologies is described. When selecting the lecture materials for this course, the introduction to the methodologies is closely combined with many industry case studies to make lecture materials easy and interesting for students. In addition, active and cooperative learning method is implemented through involving students heavily in the learning process. The implementation of this course demonstrates that combining theoretical materials and case studies, and actively involving students in the course have many positive effects on motivating students and improving the learning outcomes of this course.

Keywords: Software Engineering, Agile Development, Active and Cooperative Learning.

1. Introduction

Today's competitive market environment increasingly impacts software development projects. Changing requirements on short notice and increasing pressure to produce software at the lowest cost are inevitable. Agile software development methodologies were developed in response to such phenomenon. Contrary to traditional software development methodologies that emphasize on detailed documentations and formal processes, agile approaches share four particular values: *individuals and interactions over processes and tool; working software over comprehensive documentation; customer collaboration over contract negotiation; responding to change over following a plan*. Due to their flexibility, efficiency and cost effectiveness, agile approaches began to gain more and more acceptance among industry practitioners.

However, industry employers often voice the concerns that computer science graduates are not well prepared to function in the real world. They feel that software engineering education lags behind the industry. They complain about the slow progress to move innovations, such as agile methods, into computer science curricula.

Many universities have integrated agile software development methodologies into their software engineering curricula. However, just teaching the principles and practices of agile methodologies is not sufficient. The retention of knowledge, particular theoretical

knowledge, by students tends to be dismal: once the final exam is over, the knowledge seems to evaporate. The key skill for students to master is to understand the benefits and disadvantages of agile software development approaches, and, most importantly, when and how to use them in real world projects.

Our course on agile software development methodologies was developed in the spring of 2005. The design and implementation of the course was done jointly by two instructors from two universities in two countries: one from the Department of Mathematics and Computer Science at Salisbury University, U.S.A., and the other one from the Information Engineering Institute at Zhengzhou University, P.R. China.

In the rest of the paper, we first state the objectives for this endeavor. The detailed design and implementation of agile methodologies course are described next. Finally we analyze the course surveys and identify areas and plan for improvement.

2. Objectives

We had two major objectives when we started to design the agile software development methodologies course. The first objective concerns the content of the course. Instead of focusing on one or two well described agile approaches, we want to give students a comprehensive coverage of the field of agile software development. While introducing the principles and practices of agile methodologies, we feel that it is essential to discuss the benefits and disadvantages of these methods and incorporate real world case studies on those methodologies. This combination can make the lectures more attractive to students and teach the most important skill of when and how to successfully adopt those methodologies in real world.

Our second objective focuses on teaching method. We want to experiment active and cooperative learning approach in this course. Instead of making students passive learners, a large part of this course involves students in the learning process by having group discussions, individual in class presentations and field studies.

Besides the above two major objectives, we also hoped to improve presentation skills of the students. As mentioned earlier, this course was developed through the collaboration of two instructors from different countries. Hence we were also interested in finding out how the differences in cultural and educational background impact the learning experience in software engineering education.

3. Course Design

The design of agile software development methodologies course started in the spring of 2005. During the design process, we tried to emphasis on breadth instead of depth, combine theory and real world stories, and maximize student involvement. Next we describe the lecture content, delivery method, student involvement, and evaluation strategy for this course.

Lecture Content

During the design of agile methodologies course, we conducted extensive research on textbooks and other resource in the field. What we have found is that although there are many resources available, most of them focus on one or two well described agile approaches such as Extreme Programming only. We did find a review article [1] and it appears to be the most comprehensive review article on agile development approaches we have seen so far. As

claimed by the authors, the aim of this paper is to systematically reviewing the existing literature on agile software development methodologies. This paper proposes a definition of agile software development methodologies and a classification of agile approaches according to the definition. It also analyses ten agile software development methods, compares them and highlights their similarities and differences. Finally, future research needs are identified and discussed briefly. Those areas are very close to what we would like to cover in our course. The book “Balancing Agility and Discipline” by Boehm and Turner [4] also provides general comparisons of existing agile software development methodologies. So we decided to adopt the topics in [1] and [4] as our main content outline and supplement each topic with other materials if needed.

The course is designed for a three credit hour course for ten week period. The selected topics are divided into ten lecture units and each unit is covered per each week. The following table lists all the topics selected in this course:

Unit	Lecture Topic
1	Introduction and overview [1]
2	Extreme Programming (I) [2] <i>History, Principles, Practices, Pair Programming, Testing Driven Development</i>
3	Extreme Programming (II) [2][7][16] <i>Refactoring, Agile Project Management, The case Against XP</i>
4	Scrum [1][4][15] <i>History, processes, applications, management</i>
5	Open Source Software [13] <i>History, characteristics, impacts</i>
6	Rational Unified Process [11] <i>History, processes, features, its applications</i>
7	Other Agile Methods – Crystal Family, Feature Driven Development, Dynamic System Development, Adaptive Software [1][4] <i>History, processes, application domains</i>
8	Design in Agile Development (I) – Principles [12] <i>Single-Responsibility Principle, Open-Closed Principle, Dependency-Inversion Principle, etc.</i>
9	Design in Agile Development (II) – Patterns [12] <i>Command and Active Object, NULL Object, Façade and Mediator, etc.</i>
10	Agile Methodologies Comparisons, Experience from the Field and Future for Agile Software Development [1][4]

To supplement the content for each unit, we adopted recent research papers from ACM and IEEE journals, relevant conference proceedings, resources from the Internet, and excerpts from reference books.

To help students gain deeper and more realistic understanding of the methodologies, after introducing theoretical materials in the first lecture in each week, relevant real world

experience reports are discussed in the second lecture each week. Through this approach, we hope that students can grasp why and when to use an appropriate methodology to make a project successful. The list of real world case studies we selected contains:

No.	Case Study Topic
1	Apache and Mozilla development project [13]
2	Evolution of Microsoft software development methodology [15], http://net127.com/2005/09/24/battling-google-microsoft-changes-how-it-builds-software/
3	Refactoring examples from [7]
4	Pair programming on the C3 project [8]
5	Extreme programming in comprehensive compensation project at Chrysler Corporation http://en.wikipedia.org/wiki/Chrysler_Comprehensive_Compensation_System
6	Netscape's Navigator 3.0 project development http://www.hnehosting.com/mirrors/Origin_of_a_Browser/ http://en.wikipedia.org/wiki/Netscape#Netscape_Navigator_.28version_1.0.E2.80.934.08.29
7	Scrum development for small team [14]
8	Design principle case study – the payroll system adopted from [12]

Lecture Delivery

To achieve active and cooperative learning, weekly lecture consists of instructor lecture on methodologies and case studies, and student presentations. Specifically, the first lecture hour is for the instructor to introduce theoretical materials, the second lecture hour is for the instructor to discuss relevant case studies, and the third lecture hour is for student presentations on relevant industry experiences.

Three student presentations are scheduled during the third lecture hour. Ten minutes are given for presentation and five minutes for question and answering period. Student presentation materials are assigned by the instructor. When we assigned presentation materials, we tried to search for materials that are at appropriate level, at right length (around six pages long), and relatively new.

Student Involvement

Student involvement is an important part of this course. Besides in class presentations, students are also expected to participate in group discussions. At the end of instructor lectures, there is always a ten minute group discussion on the topic discussed that day. To wrap up group discussions, a representative from each group reports the outcome of group discussions to the whole class.

During peer presentations, the audiences are required to fill in an evaluation form. The evaluation form contains two sections. The first section is a general survey on the presentation material (such as whether the paper is interesting, new, relevant, too deep or too difficult, etc.) and what the audiences have gathered from the presentation (such as main points discussed, the audience opinion, etc). This section is designed to provide feedback on

the presentation material and what the audiences have learnt. The second session of the form is used to evaluate the performance of the speaker. It contains questions on the speaker's preparation, understanding of the material, delivery style, and presentation skills. This portion of the form will be returned to the speaker. We hope to use this approach to motivate active audience participation during student presentations.

Examination Approach

The final grade for the course is determined by in class presentation performance (20%), mid-term exam (40%) and final exam (40%).

Although the evaluation for each presentation is done by both the instructor and the audiences, it is the instructor who gives the presentation grade. The instructor grades each presentation based on enthusiasm, preparation, understanding, delivery style, effectiveness, background research, and independent thinking.

The exams cover the materials on methodologies, real world case studies and student presentations. They also contain questions on how to apply this knowledge to solve simply problems and open questions that require research outside classroom. For the final exam, each student will submit a written report on a field study for an IT industry firm or the IT department within an organization. Long before the final exam, the students are required to contact an IT organization and conduct a study on the organization. Specifically, they need to investigate the organization's main business, daily operations, revenue stream, customers, software development methodologies used, career path for IT professionals, and hiring process, etc.

4. Course Implementation

This course has been taught three times to graduate and undergraduate students at the two universities since the summer of 2005.

It was first taught at the Information Engineering Institute of Zhengzhou University in the summer of 2005. All twenty-nine students in this course are working towards their Master's degree in software engineering and are full-time employees in IT industry. In the fall of 2005, this course was taught in the Department of Mathematics and Computer Science at Salisbury University, and at the Information Engineering Institute of Zhengzhou University. The students in this course are undergraduate computer science students. The class size is eighteen and twenty-three, respectively.

To gather the students' feedback on this course, we have conducted course survey among the three groups of students. Now we summarize the survey results and discuss the issues discovered through teaching process and from the course surveys.

4.1 Survey Result Summary

The survey questionnaire focuses on the content and lecture delivery. Here are the ten questions in the questionnaire:

1. *Which topics covered in this course provide most practical value to you?*
2. *Compared with other software engineering courses you have taken before, what is uniqueness of this course in terms of delivery?*
3. *Do you feel that this delivery approach can be adopted by other courses as well?*
4. *Do you like to give in class presentations? Why? How long did it take you to prepare for your presentations?*

5. *Do you feel that this course is comprehensive enough? Why?*
6. *Do you feel that this course provide enough in-depth information of the field? Why?*
7. *What improvements does this course need in terms of it content? Why?*
8. *What improvements does this course need in terms of it delivery? Why?*
9. *Are you planning to introduce agile approaches to others?*
10. *Are you planning to adopt agile approaches in your work?*

From the survey results, we have found that the students demonstrate overwhelming positive responses to both the content and delivery approach of this course. Here is the summary of the student feedback.

The students feel that the lecture materials are up to date, relevant and interesting. They like the combination of theoretical methodologies and real world case studies. They argue that this combination make it easier for them to understand the methodologies and their applications. The real world case studies bring life to the methodologies and make the lectures more appealing to them.

Nearly all students enjoy the delivery approach. Particularly they like being actively involved in the learning process. They admit that this approach means a lot more work for them and sometimes they feel overwhelmed. But they do agree that this practice does give them opportunities to improve their presentation and independent research skills. Some students think that giving presentations is a great opportunity to communicate with people and share their understandings. One student mentioned that he had thoroughly enjoyed the experience of standing in front of an audience. Another student commented that he had never done so many presentations in his entire college life and this experience really helped him with his presentation skills.

4.2 Discussion

Through teaching this course and from the student surveys, we have also identified the following areas for improvement:

Student Participation

Although all students realize the importance of their involvement in this course, when reality hits, not all of them have made their best effort to get involved. Some still remained passive during the class. Although this may be due to personality, it was apparent that some students simply lacked the initiative and wanted to get by. For example, during group discussions in each lecture, there were always a few students sitting at the corner and not saying anything at all.

Having the students fill in an evaluation for each presentation did force their participation to certain extent, but there were always a few students who did not write down any thoughtful and useful information.

Student Presentation

We have found that the student presentations depend on many factors and the quality of the presentations varies a lot. If a student is interested in a topic and has a good attitude toward the presentation assignment, he or she usually starts to prepare for the presentation early. Besides discussion on the paper, he or she usually adds their own research results and their own thoughts to the presentation. On the contrary, if a student only treats the presentation as another assignment to earn credit, he or she tends to show very shallow understanding of the topic. The presentation, itself, is also poorly prepared (e.g., embarrassing spelling mistakes

on slides, discontinuity during presentations, etc.) and the questions from the audiences were not answered properly.

Student Background

The student background also has a lot of impacts on the learning outcome of this course. We first look at the industry background of the students. We have discovered that students with industry working experiences have deeper understanding and better appreciation of the methodologies discussed in the lectures. This is reflected by their performance during group discussions and in class presentations. On the other hand, students with little industry working experience find it less easy to see the real values of those methodologies.

The students in this course are from different cultural and educational background. When we started to design the course, a big concern for us was whether the Chinese students can understand the materials in English easily. An encouraging observation is that most Chinese students in this course have good grasp of English and have no trouble to understand scientific materials in English at all. However the active and cooperative learning approach does seem to be more challenging to the Chinese students. This may be due to the fact that they have grown up in a society where a good student should always listen and agree. They may be new to this learning approach. During student presentations, we have seen that they rarely expressed their own opinion and criticized the authors' viewpoints. The group discussions and question and answering periods were not active. However, this situation did start to improve toward the end of the course. They seemed to find the value in this learning approach and started to get used to it. On the other hand, the American students in the course are not shy to express their own opinions and provide criticisms during the presentations.

5. Future Work

To improve the above identified areas, we plan to continue our work toward the following directions:

To effectively achieve the objective for active and cooperative learning, it is mandatory to have vigorous monitoring and evaluation methods to encourage and enhance student participation. Specific criteria to track and evaluate performance in group discussions, presentations, and question and answering periods after presentations need to be developed. The performance in these activities should contribute to certain percentage of the final grade for the course.

Secondly, as we have observed, industry working experience can enhance the understanding of agile software development methodologies. We feel that it will be beneficial to incorporate small real world software development projects into this course. The students can gain first hand experience in agile software development methodologies.

Thirdly, lack of knowledge and experience on public speaking is a major cause for poor student delivery. It would be very helpful if the instructor can give a lecture on how to deliver professional presentations at the beginning of the course.

Lastly, through reading the field study reports the students submitted for the final exam, we have found that writing is another skill students need to work on. Many students have not had much experience in writing technical articles. It is hard to image what they could produce if they were asked to write a technical document by their future employers. What we have done in this course is only the first step toward improving student communication skills. The students begin to realize the importance of communication skills in this field. To continue our work in this direction, we plan to add more presentations and writing assignments in this

course so that the students can have more opportunities to practice and improve their communication skills.

6. Acknowledgement

The authors would like to thank the State of Maryland Higher Education Commission, USA, for the Welcome Fellowship.

7. References

- [1] Abrahamsson, P., Salo, O., Ronkainen, J., and J. Warsta, "Agile Software Development Method, Review and Analysis", VTT Electronic, Espoo. 107 p., VTT Publications : 478, 2002.
- [2] Beck, K., "Extreme Programming Explained – Embrace Change", Addison-Wesley, 190pp, 2000.
- [3] Bergin, J., Kussmaul, C. and T. Reichlmayr, "Agile Development in Computer Science Education: Practices and Prognosis", SIGCSE 2005, February 23-27, 2005, St. Louis, Missouri, U.S.A.
- [4] Boehm, B., and R. Turner, "Balancing Agility and Discipline", Addison-Wesley, 266pp, 2004.
- [5] Burns, J., and J. Klashner, "A Cross-Collegiate Analysis of Software Development Course Content", SIGITE'05, October 20-22, 2005, Newark, New Jersey, U.S.A.
- [6] Cockburn, A., "Agile Software Development", Pearson Education, Inc., 278pp, 2002.
- [7] Fowler, M., "Refactoring – Improving the Design of Existing Code", Addison-Wesley, 431pp, 2000.
- [8] Haungs, J., "Pair Programming on the C3 Project", Computer, February 2001, 118-119.
- [9] Hawthorne, M. and D. E. Perry, "Software Engineering Education in the Era of Outsourcing, Distributed Development, and Open Source Software: Challenges and Opportunities", ICSE'05, May 15-21, 2005, St. Louis, Missouri, U.S.A.
- [10] Highsmith, J., "Agile Project Management", Pearson Education, Inc., 277pp, 2004.
- [11] Kruchten, P., "The Rational Unified Process: An Introduction", Addison-Wesley Professional, 320pp.
- [12] Martin, R., "Agile Software Development – Principles, Patterns, and Practices", Pearson Education, Inc., 529pp, 2003.
- [13] Mockus, A., R. T. Fielding, and J.D. Herbsled, "Two Case Studies of Open Source Software Development: Apache and Mozilla", ACM Transactions on Software Engineering and Methodology, volume 11, no. 3, July 2002, 309 – 346.
- [14] Rising, L., and N. Janoff, "The Scrum Software Development Process for Small Teams", IEEE Software, July/August 2000, 26 – 32.
- [15] Schwaber, K., "Agile Project Management with Scrum", Microsoft Press, pp192, 2004.
- [16] Stephens, M., and D. Rosenberg, "Extreme Programming Refactored: The Case against XP", Apress, 400pp, 2003.