

Citace:

BUCHALCEVOVÁ, Alena. Where in the curriculum is the right place for teaching agile methods? Prague 20.08.2008 – 22.08.2008. In: Proceedings 6th ACIS International Conference on Software Engineering Research, Management & Applications (SERA 2008). Prague : Copyright, 2008, s. 205–209. ISBN 978-0-7695-3302-5.

Where in the curriculum is the right place for teaching agile methods?

Ing. Alena Buchalcevova, Ph.D

*Department of Information Technology, University of Economics
Prague, Czech Republic
buchalc@vse.cz*

Abstract

According to several agile adoption surveys made worldwide agile practices gained ground. In the Czech Republic we assumed another situation. Therefore we decided to conduct our own research based on a survey. The research objective was to determine the rate of agile approaches usage and practical experience with these approaches in companies in the Czech Republic. Paper presents the survey instrument and some results that have influenced building new Computing Curriculum involving agile methods.

1. Current state of agile development in the Czech Republic

Agile software development is an iterative process that allows small development teams to build software functionality in a collaborative environment that is responsive to business change. Development is done in short iterations (typically weeks to months) ending with working increment of software. There are many agile development methodologies, the main part of them was created prior to 2000 and named "light-weight methodologies". When the Agile Manifesto [15] was created in 2001, these methodologies, such as Extreme Programming [2], Feature-driven Development [17], Scrum [10], Crystal [9], and others were denoted as agile.

According to several agile adoption surveys made worldwide agile practices gained ground [1], [14]. In the Czech Republic we assumed another situation. Our

previous research made in 2002 exposed that the use of software development methodologies in our country was high below the world's level. We decided to conduct our own survey in 2006 as we wanted to know, whether the situation with low level of formal methodologies usage had changed with expansion of agile approaches and of course we focused the research on the Czech environment. Last but not least we wanted to examine the extent of knowledge of agile methodologies in the practical software development.

The research was carried out over a period of about six months, from December 2005 to April 2006 and was based on a questionnaire. As we assumed a limited awareness of methodologies in general, and that of agile methodologies specifically, the questionnaire was supplemented with a presentation of agile principles and the questions were formulated in such a way that they would be comprehensible to respondents with low level of knowledge of methodologies. The questionnaire was thus not only a part of the research but it also provided some basic information about agile methodologies and contributed to their publicity.

Each question contained:

- The exact wording of the question and possible answers.
- The reason why this question was included in the questionnaire.
- Information how the answer would influence the overall agility index and its calculation.

The questionnaire consisted of 18 questions and contained about 15 pages. Therefore I present only its abbreviated form:

1. Order the four parameters of a software development project (Time, Quality, Cost, Project scope) according to their priority and to how important you think it is to observe them. Please start with the most important parameter.

2. How do you deal with requirements changes?

- a. Changes are rejected
- b. All changes are under the change management
- c. Small changes are implemented, bigger - under the change mgmt
- d. Only coarse grained requirements initially, further changes are accepted

3. Do you use any software development methodology? (one of listed answers)

4. Do you adapt your methodology to individual projects (e.g. according to the project scope or criticality)? (multiple answers allowed)

- a. No adaptation.
- b. We do not adapt it because our projects have similar character
- c. We scale our manner of work for large projects to more formality
- d. We make our manner of work for small projects more flexible
- e. We adapt our manner of work according to the particular client
- f. Our methodology itself provides adaptation

5. What principles are considered crucial in your methodology? If you are not using any specific methodology what do you personally consider crucial? To each question you can attach from 1 point (not included) to 7 points (CSF for the methodology)

- a. Sequence of development phases
- b. Big design up front
- c. Sophisticated change management
- d. Sophisticated requirements management
- e. Frequent delivery of functional SW versions
- f. Source code quality
- g. Start programming as soon as possible
- h. Continuous testing
- i. Detailed project documentation
- j. Effective team communication
- k. User is integrated into development process
- l. People motivation
- m. Decision power
- n. Compliance with original requirements
- o. User satisfaction
- p. Methodology adaptation

6. What are the weaknesses of the methodology you are using? To each question you can attach from 1 point (not a problem) to 7 points (critical problem)

- a. Low level of detail
- b. Large scope, very complicated

- c. Low level of this methodology knowledge
- d. Low flexibility
- e. User is not involved
- f. Methodology dependence on technology
- g. Methodology demands high qualified people
- h. Lack of SW processes description
- i. High skilled project manager
- j. Slight stress on design quality
- k. Strong stress on design quality

7. Do you know the terms “agile methodologies” or agile approaches to software development, and do you know what they mean? (one of listed answers).

8. What do you think the advantages of agile methodologies are?

9. What do you regard as being the disadvantages of agile methodologies?

10. Assess the methodology you use in your company (if you do not use any specific methodology, assess your style of software development) according to the degree of its agility.

11. Are you considering using any specific methodology in the future (if you do not use any) or are you considering any alteration to your current methodology? (one of listed answers).

12. Assess the possible future methodology according to the degree of its agility.

13. What risks do you see in the transition to a new methodology? (multiple answers allowed).

14. What risks can you see in using agile approaches and what are your reasons for refusing them? (multiple answers allowed).

15. Agile methodologies often have more significant demands on people. Do you consider your employees' level of knowledge and readiness to agile development to be sufficient?

16. Agile methodologies often lead to a significant increase in the “creative freedom” of programmers. Do you consider your employees' character to be sufficient?

17. Based on your experience, is there any difference between Czech and foreign employees as far as the use of agile methodologies is concerned? Choose the relevant option and give a brief description of where you see any differences. If possible, give a simple example from your practice.

18. Do you consider the method and quality of software development methodologies education at Czech universities to be sufficient? If you do not, please give a brief explanation of any possible deficiencies.

The respondents (21 companies of 50 addressed) represented companies of all size, small development companies with 4–15 employees (10 companies),

middle-sized companies with 16–70 software developers (4 companies), and big companies with more than 70 developers (7 companies). 17 companies had the software development as their main activity, out of which 9 companies focused primarily on custom-made software development, 4 companies specialized in commercial off-the-shelf solutions and 4 companies developed solutions primarily for their own needs.

Description and analysis of the survey results you can find in [5]. The research has confirmed the assumption that most Czech companies do not use any public methodology. 3 of 21 respondents stated that they do not use any methodology, and these were not just small companies. 12 of 21 companies use company standards. That means they define processes, work products, templates and standards on their own, mostly inspired by some of public methodology or combination of methodologies. As for agile methodologies, Extreme programming (XP) was used in 1 company. As this company did not want to process its data anonymously, we can state its profile. This company focuses on custom-made software development in the field of internet and mobile technologies and e-commerce. Its implementation platform is Smalltalk and ASP.NET. Company uses Extreme programming in a very “clear” form, e.g. without substantial changes and uses most of XP practices.

The questionnaire included the question whether the company is considering any alteration to its existing methodology or introducing a methodology if it has not yet used any. Only in two cases did the respondents say that they were considering such an alteration. One was the “XP company”, that stated the intention to combine XP and Scrum. To sum up, companies (at least those included in the investigated sample) can hardly be expected to adopt more agile approaches.

The research confirmed also the assumption that the level of knowledge of methodologies in general and of agile methodologies specifically, is relatively limited. 5 respondents stated they had a basic knowledge of agile methodologies, 8 respondents stated the low level of knowledge, 4 respondents considered their knowledge advanced and the same number admitted this is the first time they have heard about agile methodologies. Given the fact that the respondents were carefully chosen and they had either university degree in informatics or working experience in this field, the result is rather unsatisfactory. I think one of the reasons of that fact is the language. The Czech Republic is a non native English speaking country, where knowledge of the English language especially in the older part of the population is not very good. Most methodologies

originate from English environment, books about methodologies are mostly written in English, many agile conferences take part in USA and for Czech companies and universities it is difficult to travel there. I think there is the responsibility of universities to include agile development into their computing curricula and that of companies to carry on the permanent staff education.

2. Building new curriculum with agile methods

I work at the department of Information Technologies at The Prague University of Economics. The department is in charge of lectures in informatics, where about 650 new students are accepted annually. Graduates of our Bachelor- and Master- programs are very well accepted by a wide range of companies. Our students are prepared for acting as specialists and managers for the field of informatics in various roles (IT manager, project manager, systems integrator, SW implementer, SW developer and architect, consultant, etc.).

At our department we have been facing a big challenge, when we have been building new Computing Curriculum for The European Credit Transfer System (ECTS) program in 2006. We inspired in Computing Curricula 2005 [8], IS 2002 [12] and MSIS 2000 [16] but there was no special focus on agile development in these curricula. We had some reasons for including agile topic into the curriculum:

1. I have been interested in agile methods from their emergence (I have presented them already in 2002 at the Objects conference in Prague [3]).
2. I am in the conviction that agile methods and techniques used for right projects can contribute to software project success.
3. Results of the survey I have mentioned in the first section show that not only the awareness of agile methods in the Czech Republic is rather low but there is also lack of developer skills in agile techniques.

We decided that our students should have to be taught agile approaches. But we had to resolve the question: “When in the curriculum should agile methods be introduced?” The solution is influenced by several aspects. First from my point of view students should know both approaches to software development – rigorous and agile. The main reason for that conviction is that agile methods are not suitable for every project type. In certain cases there is a need for rigorous approaches, so students should know them.

According to Scott Ambler's Agile Adoption Rate Survey performed in March 2006 [1] adoption of agile techniques (65 percent) is further ahead than adoption of agile methods (41 percent). Therefore it is more important to teach agile techniques and let students use them in programming and software engineering courses.

Based on these ideas we have defined principles for including agile topic into our new curriculum:

1. Teach agile techniques from first programming courses.
2. Use tools supporting agile techniques and methods.
3. Introduce SW development methodologies, define methodologies classification criteria, define assumptions and principles of rigorous and agile approaches, describe main representatives of rigorous and agile methodologies.
4. Establish special software engineering project course based on rigorous methodology.
5. Establish special software engineering project course based on agile methodology.
6. Use bachelor and diploma thesis for research in the field of agile approaches.

Applying these principles we have defined following curriculum structure. In the beginning of the Bachelor stage we have placed course "Programming". We teach object oriented programming in Java there. We use "Design Patterns First" approach to teaching programming [18]. We introduce Test Driven Development there from first lesson (principle 1). We use BlueJ development environment with integrated JUnit framework. This course is mandatory for all students of Informatics and is followed by another mandatory course "Software engineering". This course has two hours per week for laboratory. Students work there in teams (3- 5 people) on a small project. They intensify their Java programming and agile techniques skills particularly refactoring and Test Driven Development. They use professional IDE (Eclipse or Netbeans) integrated with Subversion version control system. For team communication they use Wiki technology (principle 2). In lectures of this course we introduce software development methodologies. We introduce methodologies classification criteria defined in [4]. We teach Multidimensional Management and Development of Information System (MMDIS), a methodology developed at the Department of Information Technology at the University of Economics in Prague [19], and Rational Unified

Process in detail. Two lectures are dedicated especially to agile methodologies. We introduce the principles of agile approaches involved in the Agile manifesto [15] and characterize main representatives of agile methodologies: Extreme programming (XP), Scrum, DSDM, FDD, ASD, Crystal, Lean development, Agile Model Driven Development. Naturally we define assumptions of rigorous and agile methodologies and teach students how to apply for each specific project the appropriate methodology.

After these two mandatory courses students have skills in agile techniques and understand purpose of software development methodologies and know main representatives (principle 3).

At University of Economics we have not special Software Engineering major but Information technology major. Only about one fifth of our major students are focused on software engineering. Therefore we offer additional software engineering education in a form of optional courses. In the field of software development methodologies we offer "Extreme programming" course and course "Object-oriented analysis and design project".

The "Extreme programming" course was first introduced in 2006. Course has two hours per week lectures and two hours per week laboratory. Lectures serve for introducing XP concepts and practices. During laboratory all 24 students are working on one project applying collective ownership of the code and other resources, short iterations, daily integration, test driven development and other XP practices.

The course "Object-oriented analysis and design project" serve as a framework for teaching various software development methodologies. The course has only laboratory lessons. Students work in a bigger team (10 people) and have the possibility to experience working on a "real" project following certain methodology. At present we teach there RUP, but for next year we plan introduce Scrum. To simulate customer participation on a project we expect that course will be led by two educators, me and a chief analyst of one software development company. She will play there a role of customer and will define requirements, assign them a priority, test and accept results.

3. Conclusions

We have just defined our new curriculum. Based on the results of our survey that have exposed low level of awareness of agile methods in the Czech Republic and lack of developer skills in agile techniques, we have decided to teach agile approaches. We have defined six

principles for including agile topic into the curriculum and applied them in defining structure and content of software engineering courses.

4. References

- [1] Ambler, S. Agile software development methods and techniques are gaining traction,
<http://www.ddj.com/dept/architect/191800169>
- [2] Beck, K. Extreme programming explained: Embrace Change, Addison-Wesley, 1999, ISBN:0201616416
- [3] Buchalcevova, A. Agile methodologies. Praha 14.11.2002 – 15.11.2002. In: Objekty 2002. Praha : ČZU, 2002, s. 53–62. ISBN 80-213-0947-4.
- [4] Buchalcevova, A.: *Methodologies for IS development and operation, categorization, agile methodologies, methodology patterns*, Grada Publishing, 2005, ISBN 80-247-1075-7
- [5] Buchalcevova, A.: Research of the Use of Agile Methodologies in the Czech Republic. Barry, C., Lang, M., Wojtkowski, W., Wojtkowski, G., Wrycza, S., & Zupancic, J. (eds) (2008) *The Inter-Networked World: ISD Theory, Practice, and Education*. Springer-Verlag: New York, ISBN 978-0387304038
- [6] Cockburn, A.: “Just-in-time methodology construction”, *Humans and Technology Technical Report* TR 2002.01, 2000
- [7] Cockburn, A.: “A Methodology Per Project”, *Humans and Technology Technical Report*, TR 99.04, 1999
- [8] Computing Curricula 2005: The overview report, 2005
<http://www.acm.org/education/curricula-recommendations>
- [9] Crystal methodologies main foyer,
http://alistair.cockburn.us/index.php/Crystal_methodologies_main_foyer
- [10] Deemer, P, Benefield, G. The Scrum Primer,
<http://www.scrumalliance.org/resources/339>
- [11] Highsmith, J.: *Agile Software Development Ecosystems*, Addison-Wesley, 2002, ISBN 0-201-76043-6
- [12] IS 2002: Curriculum Guidelines for Undergraduate Degree Programs in Information Systems,
<http://www.acm.org/education/curricula-recommendations>
- [13] Kadlec, V.: *Agile programming*, Computer Press, 2004, ISBN 80-251-0342-0
- [14] Larsen, D. Agile Alliance Survey: Are We There Yet?
<http://www.infoq.com/articles/agile-alliance-survey-2006>
- [15] Manifesto for Agile Software Development
<http://agilemanifesto.org/>
- [16] MSIS 2000: Model Curriculum and Guidelines for Graduate Degree Programs in Information Systems
<http://www.acm.org/education/curricula-recommendations>
- [17] Palmer, S.R, Felsing, J.M.: *A Practical Guide to Feature-Driven Development*, PrenticeHall, 2002, ISBN 0-13-067615-2
- [18] Pavlickova, J., Pavlicek, L, Pecinovsky, R. Let's Modify the Objects-First Approach into Design-First. Bologna 26.06.2006 – 28.06.2006. In: ITICSE06 [CD-ROM]. New York : ACM, 2006, s. 188–192. ISBN 1-59593-346-8.
- [19] Vorisek, J., *Strategic management of information systems and systems integration*. Management Press, Prague 1997, ISBN 80-85943-40-9