# The Modern Educational Course on Agile Software Development

Evgeny Sorokin
Lobachevsky State University of Nizhni Novgorod,
Russia
e-mail: evgeny.sorokin@inbox.ru

Kirill Kornyakov
Lobachevsky State University of Nizhni Novgorod,
Russia
e-mail: kirill.kornyakov@gmail.com

***Abstract* — the article presents new educational course dedicated to Agile Software Development. The course consists of theoretical and practical parts. Theoretical part gives an overview of classical agile methodologies, widely accepted by industry practices and some important programming principles and patterns. The second part of the course includes several labs, aimed for practical usage of agile development principles and patterns. Major feature of the course is its practical orientation and focus on agile ideas accepted by the industry.**

***Keywords – agile; software development; software engineering education.***

## I. Introduction

Agile methodologies are a trend nowadays. The most appropriate way to work in startup companies is to adopt software process, which makes team responsive to market. The same situation is observed within large companies sometimes [1, 2, 3]. Heavy software processes are usually good for long-term projects with well-defined requirements. Small projects and teams prefer lightweight techniques, which can help them work effectively in uncertain conditions.

It is most likely that young software engineer will participate in an agile software project. Regardless of this, every developer need to now basic agile development principles and practices, which help to create high-quality, maintainable and flexible software systems. For example, *refactoring* and *continuous integration* practices are de-facto standard for now, SOLID object-oriented design (OOD) principles are also very important for quality of software.

We have created a modern Agile Development educational course. The goal of the course is to create a set of materials suitable for training students (and developers) on agile concepts and practices. This is the list of major course features:

- General overview of Agile methodologies;
- Detailed discussion of the Scrum and eXtreme Programming (XP) methodologies;
- Overview of architectural styles;
- Introduction to Domain-Driven Design (DDD);
- Practical explanation of SOLID principles;
- Practical exercises on Test-Driven Development (TDD) technique.

The rest of the paper is organized as follows. In section 2 we discuss the purpose of the course. Section 3 and 4 give course overview and information about course evaluation. Benefits to the Software Engineering community are discussed in section 5, section 6 concludes the paper.

## II. Course Purpose

Agile methodologies are a growing trend for software companies nowadays. Many and many projects use Scrum, XP, Kanban etc. But there is no complete course in academic program, which covers agile methodologies in such a way so that student can learn key principles and practices of that domain. At present time we can see numerous materials about Software Engineering (SE) and Agile Development particularly. Most of them can be subdivided into the following categories:

- Classical Software Engineering courses [4, 5, 6].
- Pioneering books about Agile Development [7, 8, 9] and their followings.
- Commercial training courses by consulting companies for developers and managers.
- Extensive research on applicability of agile practices in various environments.

We can see strong attention to the Agile methods from both industry (software development and consulting companies) and academy. But there is still a lack of free Agile Development courses with strong practical orientation. There is now way for beginner developer to understand agile programming practices except joining an agile team.

Such situation results in mutual disappointments of employer and employee during the first years of software engineer’s career. It also results in a number of failures in startup projects because of the ignorance of ideas and recommendation of agile evangelists.

Another point is that classic publications about Agile don't reflect corrections made by industrial usage and patterns of adaptation. There are lots of materials in the Internet, which explains concepts of Agile but there are no free stand-alone workshops aimed for complete overview of

Agile world and strong practicing of agile techniques in software projects.

So, there is a demand for a modern and open course, which can be used for training students and software developers.

## III. Course Description

To simplify and facilitate the entering into industrial software development the agile software development course has been elaborated. Below we combine the description of the course and the experience report of its conduction. The important point is that we encourage senior engineers from Nizhni Novgorod Hi Tech companies to participate in that course as guest lecturers. This year we have invited IT specialists from Intel [10] and Itseez [11], who have explained materials from their day-to-day work. This has a serious value to the audience because it, on the one hand, demonstrates that the material is being applied in the real world and, on the other hand, exposes the tactical and practical aspects.

The course provides complete overview of the agile software development world: Agile Manifesto, detailed description of the most authentic methodologies — Scrum and XP, brief introduction to software adaptation of Kanban (Toyota Production System) and Lean philosophy.

The second part of the course contains introduction to software architecture and design. Again, the course approaches this topic from the point of practice: instead of UML diagrams lectures demonstrate the examples of real-life software and/or scalable approaches of architectural decision making. The specifics of the materials chosen from the Software Engineering are based on real industry demand. Many IT companies in Russia provide outsourcing of business application development. So as authors of that course we made the analysis of most popular practices adopted by the IT community. We have conducted several interviews with representatives of different software companies of Nizhni Novgorod to define their needs.

The stories about success and failures of real adoption of agile methodologies from practitioners are supported by set of handouts and homework. The practical exercises are dedicated to the different programming practices (refactoring, Test-Driven Development (TDD), SOLID principles of OOD, application patterns, Domain-Driven Design (DDD)).

DDD is a modern approach to software development by a structure of practices and terminology. DDD helps to make architectural and design decisions for software projects dealing with complex domains.

TDD is a software development technique which requires automated unit test to be written *before* producing the code which passes that test. This technique influences design solutions (makes them simple and flexible) and provides validation of correctness.

The practical exercises helped students to feel deeply do's and don'ts of practical coding of application in uncertain conditions. The requirements to the lab work were general and amended based on concrete solution with intent — to practice refactoring to patterns, to let students see the value of TDD and to educate dependency management.

Another point is that students used the same tools, which usually being used in software companies. The labs were created in Microsoft Visual Studio with C# programming language. For refactoring audience used ReSharper from JetBrains company [12]. It matters because it helped audience to increase their productivity and leave mechanical routine of working with syntax to the tools. It is extremely important for adopting TDD for example.

The tasks of lab work were designed in the way that students could complete the product. For example, as a result of TDD task students have DLL, which implements something useful and a set of unit tests, which supports the implementation. As a result of lab on Passive View pattern [13] students have complete Windows- or Web- application with architectural layers and another set of unit tests for validation of the presentation logic.

## IV. Course Evaluation

Currently we have a strong support from lecturers, engineers and managers from UNN [16], Itseez, Intel, ITLab [14].

- Nizhni Novgorod State University facilities are used as a basis for initial course approbations. The course has been conducted at the Master's course of Faculty of Computational Mathematics and Cybernetics of State University of Nizhni Novgorod.
- The course will be tested at Winter and Summer Schools (annual sessions) at ITLab.
- Members of ITLab Seminar on Software Engineering [15] offer their help for course materials review and discussions.
- Several IT companies of Nizhni Novgorod suggested their free consulting services and opportunity to test effectiveness of course materials for their beginner developers training.

Among the results of the course we can emphasize the following:

- Students acknowledge and appreciate the part of the course, which covers process practices. Examination has demonstrated that key concepts of agile software development were easy to perceive.
- TDD lab work has been the one of the most "heavy to learn" practices. Students have often considered TDD as just writing of unit tests.
- Students who have applied all lab works to a single software project appeared higher results than those who have created new projects for every lab work.
- The materials of the engineering part of the course, which haven't been covered by lab work (such as DDD) were not clear to the audience.

## V. Benefits to the Software Engineering Community

The course brings the following benefits to the SE community:

- Course materials will be available online for free. Community will get the tool for educating students and developers for agile concepts and techniques.
- Graduates will have strong agile development knowledge, adequate skills for fast integration into software development process, ability to adopt agile methodologies and build up a unique software process for small and medium sized software projects. We expect to see developers with deep understanding of what Agile is, and what it's not. We hope that students will be ready for joining agile teams and initiating their own agile projects.
- This course will be valuable for the whole Russian-speaking software development community. Especially because there is very few information on Agile in Russian language.

## VI. Conclusion

Initial version of modern Agile Development course have been created and evaluated as Master's course at the State University of Nizhni Novgorod, Faculty of Computational Mathematics and Cybernetics. Feedback was positive in general both from students and evaluating experts.

We are looking forward to hear from all whom this course may concern, especially lecturers who also would like to develop this course together with us, IT companies who would like to conduct it for their employees (or to participate as guest lecturers) and for anyone who would like to review our course and help us to improve it.

## References

[1] The Dancing Agile Elephant: IBM Software Group's Transition to Agile and Lean Development http://www.infoq.com/presentations/dancing-agile-elephant

[2] Agile adoption at Google: Potential and challenges of a true bottom-up organization http://www.agile2007.org/agile2007/index.php%3Fpage=sub%252F&id=713.html

[3] The Microsoft Solutions Framework: An Integrated Approach to Agile or Formal Software Development Process http://blogs.msdn.com/askburton/articles/330974.aspx

[4] Ian Sommerville, Software Engineering, 8 ed., Addison Wesley, 2006.

[5] R. Pressman, Software Engineering: A Practitioner's Approach,McGraw-Hill, 2001.

[6] S. McConnell, Rapid Development: Taming Wild Software Schedules, Microsoft Press Books, 1996.

[7] A. Cockburn, Agile Software Development. Reading, Massachusetts: Addison Wesley Longman, 2001.

[8] K. Schwaber and M. Beedle, Agile Software Development with SCRUM. Prentice-Hall, 2002.

[9] K. Beck, Extreme Programming Explained: Embrace Change, Second ed. Reading, Massachusetts: Addison-Wesley, 2005.

[10] http://www.intel.com

[11] http://www.itseez.com

[12] http://www.jetbrains.com

[13] http://martinfowler.com/eaaDev/PassiveScreen.html

[14] http://itlab.unn.ru

[15] http://groups.google.ru/group/itlab_se?hl=ru

[16] http://unn.ru