

Use of Agile Methodology in Development of Educational Software for Users with Specific Disorders

Rostislav Fojtik

Abstract: *The paper shows suitability of using agile methods of software development for a concrete application which is designed for users with specific disorders, such as autistic spectrum disorders, Down syndrome, mental retardation, etc. It tries to show advantages and disadvantages of new methodologies, particularly Extreme Programming. Agile methodologies of software development appeared in the second half of the 90's of the last century. Thus it concerns new ways which have not been spread massively yet. However, they are perceived as an alternative and successful way of software development. They have been increasingly applied in smaller teams. Many of agile processes and techniques influence traditional rigorous methodologies.*

Key words: *Agile Methodology, Autism, Down Syndrome, Extreme Programming, Mental Retardation, VOKS, RUP, Testing*

INTRODUCTION

Currently, we can meet ad hoc software development which is not based on any methodology and the process is controlled rather intuitively. This approach is, however, possible to be used only in small and simple projects. Extending requirements on software and speed of its development have led to the necessity of more sophisticated development approaches. Individual methodologies and models have gradually appeared. Each of them has its pros and contras. All of them try to create and describe the way to develop software most effectively.

Historically, among the first models can be classified the waterfall model, which is described as a sequential process with individual stages strictly separated and access to the next phase is possible after completion of the proceeding one. Currently, this model is used less and less. At the time of its creation (1970), it meant a breakthrough in the view on software development and it was the first model. Its basic characteristic is sequentiality of individual stages:

1. Problem definition.
2. Analysis of requirements.
3. System proposal.
4. System implementation.
5. System integration and testing.
6. Operation and maintenance.

The advantages of this model include simplicity, sufficient familiarity among developers, ideal conditions for control. The model is easily understandable, the team members understand well their role in the model. On the other hand, crucial disadvantages count in inflexibility because software delivery is done in a form of a big bang. In fact, the customer is not involved in the development process. Advantage and disadvantage at the same time is an emphasis on the assignment. It can happen that an assignment misunderstood by the analyst is not revealed until testing or at the moment of transition [8].

Disadvantages of the waterfall model might be eliminated by the spiral model of software development lifecycle, which introduces two breaking concepts:

- iterative approach,
- repeated risk analysis.

Currently, software development companies use RUP methodology. It concerns commercial methodology created by Rational Software Corporation, currently an IBM division. The methodology is object-oriented, whose basic element is a use-case-driven approach. The development proceeds in iterations, each having four stages:

- Inception
- Elaboration
- Construction
- Transition

Strong sides of the methodology are: generality, range, robustness. Therefore it is suitable for a wide range of projects. All steps are described and documented in detail. All activities have available templates, guides, examples, and sophisticated tools. It concerns a complex methodology covering the whole lifecycle support and concrete solutions of all activities during software development. It is primarily suitable for larger teams and extensive projects. It requires a precisely documented and set formal development process, which is directive-controlled to a large extent. The methodology can adapt individual needs, but it requires an experienced project manager and plenty of time. RUP is based on an object-oriented approach. There are also strong ties and interconnections with various CASE and other supportive tools, from IBM or others. The necessity of profound knowledge of the methodology is its disadvantage. New teams must count with a long time before getting acquainted with the methodology itself.

Unified Software Development Process, Unified Process (UP) in short, stems from commercial methodology RUP. It also contains four basic stages, yet there are no supportive tools and other support as in commercial RUP.

AGILE METHODOLOGIES

Current software development is characterised by shorter and shorter lifecycle. In addition, the development reflects running changes and dynamic technological development. It can be seen not only when developing complex and extensive information systems, but when developing specialised applications as well. Traditional software development methodologies do not meet current conditions; therefore those are agile methodologies that have taken up their place in recent years. Despite a certain difference, they have similar principles, which were expressed in Agile Manifesto in 2001 [7]:

- Individuals and interactions over processes and tools
- Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change over following a plan

Apart from extensive projects with clear and detailed assignment solved by large development teams, there are also small projects. Their development is secured by a small group of developers. In many of these projects, we cannot pre-define all requirements for the application, or the customer cannot do it. An example can be specialised applications for a targeted group of end-users. We could class here educational software for users with specific disorders.

Recent years have proved that so-called agile methods of software development can be used for this purpose. They were created on the basis that traditional rigorous methodologies are no longer suitable due to their formality and hugeness. It typically concerns web applications and closely specialised software. Customers are not usually willing to wait for months for a web application to be created. They prefer fast implementation even at the cost of continuously created modules [1, 3].

Basic requirements on agile methodology cover:

- development is controlled by current requirements on functionality
- emphasis on continuous communication between the development team and customer
- emphasis on teamwork and team self-organisation
- regular and frequent transition of completed work to the customer
- not to avoid changes in the programme

- emphasis on the output programme quality before documentation
- changes should be actively followed and commented by the customer
- customer can actively interfere into the development

Agile methodologies include:

- Dynamic Systems Development Method (DSDM),
- Adaptive Software Development (ASD),
- Feature–Driven Development (FDD),
- Extreme Programming (XP),
- Lean Development,
- Scrum,
- Crystal Methodology,
- Agile Modeling.

Table 1. Comparison of basic differences between traditional and agile methodologies

Agile methodologies	Traditional methodologies
Requirements on the application change frequently	Requirement do not change in the course of the development, or only minimally
Principle of freer cooperation of the development team	The development principle is based on fixed order
Developers should have experience needed for process adaptation	Less experienced developers
Emphasis on team communication	Directive management
High tolerance to changes	Low tolerance to changes
Testing throughout the whole course of the development	Testing is usually done at he end of the development
Documentation and models do not play the most important part in the development process	Documentation and the created model have high importance and the developers must follow them
The customer must be a part of the team	Customer's role is reduced to only the starting and final stages of the development
Smaller teams (2 to 10 developers)	Designed rather for larger teams

Agile methodologies are not suitable for all types of projects and all types of developing teams. Therefore there is a frequent combination of these two approaches. As for example, we can take methodology RUP, which gradually incorporates lots of agile techniques. RUP is currently representing a kind of a framework including both traditional and agile methodologies.

Despite the fact that the use of agile methodologies is increasing, their expansion is not still common. For example, 66% out of 3061 respondents of an inquiry held in 2008 said that they used at least some of agile procedures, such as iteration planning, unit testing, daily standup, release planning, continuous integration [10]. The most frequently accepted methodologies are XP, Scrum, Agile MSF, Agile Unified Process. On the other hand, according to inquires made in the Czech Republic, the situation is much worse. Only 43% of the respondents are aware of agile methodologies, but majority of companies do not use these methodologies at all [2].

AGILE DEVELOPMENT OF VOKS APPLICATION

The Czech system VOKS mainly stems from the methodics and principles of PECS (The Picture Exchange Communication System). The base is a use of pictures representing particular concepts, things, activities. It is an alternative education and communication method for communication between an adult and a child handicapped by autism, mental retardation, and other specific disorders [5, 9]. This application requires two basic parts. The first one – a functional part designed for children. The second serves to

their parents or health workers for configuration and setting. Unlike common applications, requirements on functionality and look of an application for particular individuals are often very distinct and influenced by health or mental problems of the child.

Initial efforts in the development stemming from the waterfall model have proved unusable. The customer (health staff of special school) did not have a clear idea about all functionalities and qualities of the developed communication software. Lots of the requirements had to be corrected, specified, and re-defined. Thus we proceeded to change the strategy and the software was developed according to an agile methodology. Elements and processes of Extreme Programming were taken as the basis. The basis of the whole XP software development is code-writing and testing. XP methodology is primarily designed for smaller teams with two to ten members, who work on frequently or less known assignments. Projects which take long or have difficulties in getting feedback (e.g. from technological point of view) are not suitable for this methodology. Automatic testing or version assembly is necessary for its implementation. XP is a flexible agile methodology emphasising interconnection of the proposal and implementation stages [6].

Basic activities are (Fig.1):

1. Planning and Managing
2. Designing
3. Coding
4. Testing

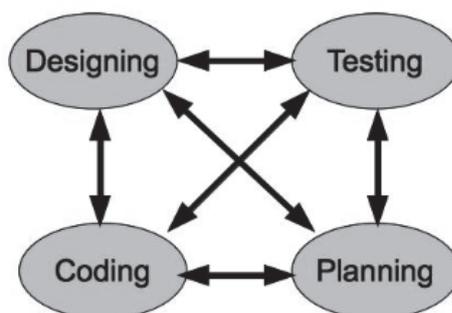


Fig.1. Basic activities of Extreme Programming

XP has the following characteristic qualities:

- Continuous revision of the program code – frequent use of pair programming, when a pair of programmers works together on one code. Application of the principle led to removal of previously occurring problems in the code of the developed application.
- Testing – apart from unit tests, continuous testing by the customer was also successful. In our case, it was primarily health staff, but also children with specific disorders. Only the health staff from special school could define whether the application was suitable.
- Short iterations – classic methodologies usually transit the application to the customer after a long period, usually at the end of the development. This approach in the development of the educational application did not prove to be suitable. When an agile development methodology started to be used, we tried to make the shortest and most frequent iterations. They were given by time possibilities of the participants and we can see from the acquired experience that it would be convenient to make iterations more often than have been made so far.

Extreme Programming consists in five values:

- *Communication* – a large number of development problems lie in incorrect communication, not only among the team members, but with the customer as well. If XP is used, large teams assign a special role, so-called coach, who detects communication failures and secures correct communication. However, small teams should not

underestimate good, continuous communication. Frequent communication is important not only among the developers, but with the customer as well.

- *Simplicity* – the methodology tries to develop software as easy as possible, not to deal with functionalities that are not currently important and that might be used in future. XP methodology says we should not create a more robust architecture than necessary for the moment. It also proved crucial in the development of the PECS application. With respect to frequently unclear customer's requirements, it was the most effective to solve the application for future needs. The development required more time and energy and certain functionalities frequently proved to be unnecessary or unusable for a specific group of users.
- *Feedback* – is very important for correct development. It runs at several levels. One of them is testing, which should be performed at all development stages and not after the implementation stage. When developing program PECS, it proved necessary to test it continuously by its users. The main reason was the need of the developers to become familiar with possibilities of users with specific disorders. Many of commonly used approaches (particularly in the area of program control) are not usable for this group of users. Thus it was necessary to test more frequently and in all stages of the development what suits the users or what does not. The second target group of users was the health staff of special school that had to define and set up in the configuration part of the program an individual educational program for individuals. The configuration program requires simple control, still it offers a wide range of possibilities of individual setting according to specific requirement of the tutored child.
- *Courage* – A very important value of XP is courage to correct and remove errors at all costs. It even means removing a great part of the code or fundamental re-doing of the so-far architecture design. According to experience with the practical use of XP in companies, it seems that this requirement is difficult to be applied (more difficult in Europe than in the USA). The developers feel that removal of a great part of the code signs their failure and they are less likely to try further [8]. In the case of the PECS application development, it was necessary to remove a part of already created code several times because the corresponding application functionality proved to be unsuitable for a user with specific disorders. Despite those fundamental changes, it appears that the process helps to achieve the objective better than traditional methods.
- *Respect* – The team members should be interested in their colleagues' work. In case individuals work alone, with no close relations to others, XP will be unusable. This XP value closely corresponds to the emphasis on communication.

Why is agile methodology suitable for developing educational and communication software for users with specific disorders?

- The submitter does not have a clear idea about functionality of the application. Because there is a lack of experience with similar software, the development is frequently changed or corrected.
- A closely specialised group of future users requires frequent communication with submitters as well as continuous testing on by the developers and future users.
- The number of developed similar applications is very small; therefore there is a lack of experience with similar projects. Traditional development processes cannot be used in many aspects, particularly in the area of program control.
- The developers must get thoroughly acquainted with the environment and users, with whom they do not often get in touch and thus they do not know their abilities and requirements.

The used methodology has its disadvantages and risks. The requirement on frequent iterations proved to be hardly realisable. With respect to the nature of the solved project,

direct communication of the developing team and the staff of the special school was necessary. It was quite a difficult task to secure this way of communication (taking into consideration time possibilities and distances between the participants). In order to achieve faster advance in the development of educational software, it would be necessary to communicate in shorter intervals, which was not possible to carry out successfully.

CONCLUSIONS

Extreme Programming and other agile development methods have been implemented more and more in the past decade. It does not concern methodologies that would be suitable for all projects, they find their place in smaller developing teams. They can be used in projects with no clear initial definition, where the customers do not have a clear idea about the output product. Extreme Programming does not emphasise documentation of the development and its strict control. The basic element is a high level of communication among all team members and customers as well as frequent iterations. An advantage of this methodology was its possibility to react fast on customer's changes in requirements and possibility to adapt the program to users with specific disorders, even at the cost of removal of a great part of already written code. The methodology prefers fast reaction to a change before the plan completion, which proved to be important in the case of the developed software. There were plenty of changes and new requirements during the development, and their solution was more important for the output quality than following the time schedule of the development.

REFERENCES

- [1] AMBLER, S. Agile software development methods and techniques are gain traction, 2006, <http://www.it-smc.com/Articles/Survey%20Says%20-%20Agile%20Works%20in%20Practice.pdf> [on-line]
- [2] BUCHALCEVOVÁ, A. Agilní metody, jak dál? *Tvorba softwaru 2008*. VŠB TU EF, Ostrava 2008, s. 30–34.
- [3] BUCHALCEVOVÁ, A. Selection of the Appropriate Methodology for Concrete Project. *Objekty 2009*. Univerzita Hradec Králové, 2009, s. 37-48.
- [4] BUCHALCEVOVÁ, A., KUCHERA, J. Hodnocení metodik vývoje informacních systému z pohledu testování. *Systémová integrace*, 2008, roc. 15, c. 2, s. 42-54.
- [5] CHARLOP-CHRISTY M.H., CARPENTER M, LE L., LeBLANC L, KELLEY K. Using the Picture Exchange Communication System (PECS) with children with autism: Assessment of PECS acquisition, speech, social-communicative behavior, and problem behaviors. *Journal of Applied Behavior Analysis*, 35, 213-231, 2002
- [6] COLLINS, Ch. T., Miller R. W. Adaption: XP Style, 2001, http://www.christophertcollins.com/papers/adaptationXpStyle_final.pdf [on-line]
- [7] FOWLER, M. The Agile Manifesto: where it came from and where it may go, <http://martinfowler.com/articles/agileStory.html> [on-line]
- [8] KADLEC, V. *Agilní programování*, Computer Press, Brno 2004.
- [9] KNAPCOVÁ M. *Výměnný obrázkový komunikační systém - VOKS*. Praha: Institut pedagogiko – psychologického poradenství ČR, 2005.
- [10] The State of Agile Development, 2008, http://www.versionone.com/pdf/3rdAnnualStateOfAgile_FullDataReport.pdf [on-line]

ABOUT THE AUTHOR

Rostislav Fojtik, PhD, Department of Computer Science, Faculty of Science, University of Ostrava, Czech Republic, 30.dubna 22, Ostrava, rostislav.fojtik@osu.cz, Phone: +420597092173