Proceedings of the International Multiconference on
Computer Science and Information Technology pp. 687–691
ISBN 978-83-60810-27-9
ISSN 1896-7094

Agile methodology and development of software for users with specific disorders

Rostislav Fojtik
University of Ostrava, Department
of Computer Science, 30.dubna
22, Ostrava, Czech republic
Email: rostislav.fojtik@osu.cz

***Abstract*—The paper deals with possibilities of information technologies when improving communicative skills of children with specific disorders, such as autistic spectrum disorders, Down syndrome, mental retardation, etc. The development of an application stemming from the communication system PECS (The Picture Exchange Communication System) and its Czech variant VOKS is the base of this paper to show specificity of the development and verification of software for the given group of handicapped users. The paper shows suitability of using agile methods of software development for a concrete application which is designed for users with specific disorders. 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.**

***Key words*: Agile methodology, autism, Down syndrome, Extreme Programming, mental retardation, VOKS, RUP, testing**

I. Introduction

USE OF information and communication technologies has reached massive expansion recently. Computer users are to be found in every social and professional group. Computers simplify a lot of work tasks, make methods of communication easier and more accessible, intermediate new information, etc. These modern technologies can play a role of important supporting and compensational means for a group of users with specific needs. It is mainly people with some mental and health problems who face difficulties with communication, and common means of communication are usually unusable for them. Computers can act as a positive agent in such cases. However, hardware and software must be adjusted to the specific needs and it is not possible to implement experience gained and applied with common groups of users.

There is a potential group of users with specific disorders who could make use of the development of communication skills. E.g. people with autistic spectrum disorders, Down syndrome, forms of mental retardation and other development defects which cause problems to learn and use spoken language, written text, and other commonly used means of communication. Various methodics bringing interesting results have been worked out to develop communication skills of people with the above mentioned disorders. For instance, such methodics is the system PECS (The Picture Exchange Communication System) and its Czech variation VOKS. The base is a use of pictures representing particular concepts, things, activities. The user gradually becomes familiar with new concepts – pictures, which are then incorporated in their list. Choosing pictures and placing them on a sentence strip helps them learn to create even simple sentences. [7]

II. Metodics for learning

Based on the requirements from special school, we are developing an application which should facilitate communication with a child with specific disorders such as autism, Down syndrome, mental retardation and others. Autism is a mental handicap which demonstrates in worsened communication skills. Speech formation of people with an autistic spectrum disorder is delayed and some of them do not speak all their life or they use only limited number of words, frequently corrupted. Several methodics to improve their communication skills have been developed. [15]

One of them is the PECS methodics created in 1985. 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. [16] The fundamental principle is an exchange of a picture representing a particular thing for a real one. Continuous practice leads the child to learn how to use concepts correctly and improve its communication skills. The whole process is emphasised by stimulation. For instance, the child gets a chocolate if it brings the correct picture. Follow-up education leads not only to knowledge of pictures – concepts, but also to composing a short sentence from the pictures. The last phase of education encourages the children to comment independently the action around them and to answer direct questions. The system also helps some preschool children to develop speech. An elaborate methodics of work with a system provides six training phases with examples, manuals, and advice [3] [2].

The Czech system VOKS mainly stems from the methodics and principles of PECS. However, it brings some improvements and it tries to adjust to particularities of the Czech language. Unlike PECS, it emphasises visual support of speech of both communicating partners. Thus there are changes to basic situations in all lessons. The methodics is

divided into two basic parts. The first contains important information concerning teachers of the VOKS system and training environment. In addition, it describes preparation of individual tools for communication and the way of reward choice before the whole training of the communication begins. The second part contains educative lessons, which form the backbone of the education. The client learns to ask spontaneously for a favourite thing in exchange for a picture, to go on his own to the symbol container to get a picture and then to hand it in subsequently to the partner. They learn to ask different people about anything in an unfamiliar environment, to choose appropriate symbol from several pictures, to complete correctly a simple sentence on a sentence strip using pictures and then to ask by help of the sentence strip. Finally, they learn to react to various questions and to comment on the environment around them. The supplementary lessons then develop picture inventory and syntax [13].

III. Application voks

When designing a new application environment, it was necessary to get rid of all control features which were not directly related to the main use of the programme, and which could lead to early closing, new configuration, initialisation of new actions, etc. A lot of children find difficult to concentrate on realizing the needed performance. They frequently unwillingly click the mouse or wander the cursor round the screen. Thus it is highly important that children could not consciously or by mistake initiate functions indirectly related to the course of the programme.

The application control must not be demanding, unclear, or complicated at all. Generally common activities (such as double-click on a mouse) are almost impossible for numerous users with specific disorders. The control features of the programme must be conveniently large and their start-up and control as easy as possible. An ideal case would be if the application, having been opened, took up the whole screen of the desktop, being maximized all the time with no possibility of any change. The use of the mentioned software counts with the use of special hardware to control the computer. Users with motor handicap are e.g. supplied with IntegraSwitch, which works as an aspiration-expiration switch. Classical mice and keyboards are replaced by alternative positioning devices and keyboards, such as BickTrack, KidTrack, Roller Joyistick, BigKeys, IntelliKeys, various sensor buttons, and others. The main advantages are larger control features and more robust construction. On the other hand, the disadvantage is usually the price which exceeds several times the price of commonly used hardware. Purchase of such special devices can become unaffordable for many families with handicapped children. The solution could be use of touch screens, e.g. in new types of computers generally called nettops. It is a kind of cheaper computers containing all parts in one case together with a touch screen. The user can control the applications directly on the touch screen, which seems to be more convenient especially for children with specific disorders than using mouse or other positioning devices. Some children have motor problems when using mouse, moreover they have problems with moving the mouse while concentrating on another place - on the screen. Nevertheless, despite all of the above mentioned special hardware tools, it is convenient to create the developed software in a way to be usable even on common desktops, or notebooks with their usual peripherals.

IV. 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 [9]

- 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] [5]

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 the 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. [12] 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. [4]

V. AGILE DEVELOPMENT OF VOKS APPLICATION

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.

Basic activities are

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

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. [12] 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 n shorter intervals, which was not possible to carry out successfully.

VI. Testing of voks application

Hand in hand with an increase in requirements and properties of software tools, there is a need of appropriate and profound testing. Despite the fact that the above described programme does not include any complicated functional structures or algorithms, and work of complicated and expensive devices is independent of its activity, it is necessary to secure its high reliability. Children suffering from e.g. autism find difficult to get used to a new environment and unfamiliar things. The time needed for the child to accept the programme and learn to work with it is usually very long. In case of serious disorders, there is a danger that the child will decline it or refuse to work with it. During testing of an application by children, there should not be any fundamental adjustments of the graphical interface and its control. Apart from classical procedures of testing of the programme functionality, it is important to observe how the child masters the application. It is not enough to use only usual static and dynamic testing means, analysis of the source code, monitoring by testing programme, special tests on memory usage or load, etc. [6]
Thus the application incorporates internal mechanisms which monitor user's activities. Monitoring data about manipulation with programme objects are stored in XML files. The application contains an interface designed purely for parents and health workers which enables to evaluate both child's skills to work with the programme and its functionality. The output data then acts as a feedback for the creators of the programme as well as for parents and health workers, who can adjust the process of education more effectively. Another way to verify the programme functionality is methods used in quality oriented pedagogical research. It primarily concerns the method of observation [10]. The children use the application together with the parents or health workers. The adults act both as a pedagogue, who teaches the child to use and communicate with the programme, and an observer, who check for reactions and skills of the child. The observation results are very important not only for the educational process, but also for the development of the programme.
Unlike commonly used software, there are specific problems when testing:

- The target group of users is not large enough.
- The target group of users is very diverse and the level of the disorder considerably influences ability to work with the programme. It is necessary to take into consideration individual needs of the users and to enable more possibilities of setting and adjustment.

- The phase of learning how to use the programme is very time consuming. This results in longer period of testing than with usual applications. Children with higher level of disorder can take months to pass from one phase of the programme to another.

It is important to largely cooperate with parents and health workers because the children find difficult to get used to changes in environment and unknown people.

VII. 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] Begeer S., Banerjee R., Lunenburg P. Brief Report: Self-Presentation of Children with Autism Spectrum Disorders, *Journal of Autism and Developmental Disorders*, Volume 38, Number 6 / July, 2008, ISSN 1573-3432

[3] Bondy S. A., Frost L. *What is PECS?*, http://www.pecs.com/WhatsPECS.htm

[4] Buchalcevová A. Agilní metody, jak dál? *Tvorba softwaru 2008*. VŠB TU EF, Ostrava 2008, s. 30–34. ISBN 978-80-248-1765-1.

[5] Buchalcevová A. Selection of the Appropriate Methodology for Concrete Project. In: *Objekty 2009*. Univerzita Hradec Králové, 2009, s. 37–48. ISBN 978-80-7435-009-2

[6] Buchalcevová A, Kučera J. Hodnocení metodik vývoje informačních systémů z pohledu testování. *Systémová integrace*, 2008, roč. 15, č. 2, s. 42–54. ISSN 1210-9479

[7] 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, http://seab.envmed.rochester.edu/jaba/articles/2002/jaba-35-03-0213.pdf\

[8] Collins Ch. T., Miller R. W. Adaption: XP Style, 2001, http://www.christophertcollins.com/papers/adaptationXpStyle_final.pdf [on-line]

[9] Fowler M. The Agile Manifesto: where it came from and where it may go, http://martinfowler.com/articles/agileStory.html [on-line]

[10] Gavora P. *Úvod do pedagogického výzkumu*, Paido, Brno 2000, ISBN 80-85931-796

[11] Hoekstra R. A., Bartels M., Cath C. D., Boomsma D. I. Factor Structure, Reliability and Criterion Validity of the Autism-Spectrum Quotient (AQ): A Study in Dutch Population and Patient Groups, *Journal of Autism and Developmental Disorders*, Volume 38, Number 8 / September, 2008, ISSN 1573-3432, http://www.springerlink.com/content/m184028q8305/?sortorder=asc&p_o=10

[12] Kadlec V. *Agilní programování*, Computer Press, Brno 2004, ISBN 80-251-0342-0

[13] Knapcová M. *Výměnný obrázkový komunikační systém - VOKS*. Praha: Institut pedagogiko – psychologického poradenství ČR, 2005, ISBN 80 – 86856 – 07 – 0

[14] Scherz J., Meng-Ju T., Broston S. *Adult Preferences Between Two Symbol Sets: Comparing Boardmaker and Overboard*, Wichita State University 2006, http://convention.asha.org/2006/handouts/855_1293Scherz_Julie_073114_111306111037.pdf

[15] http://en.wikipedia.org/wiki/Autism

[16] http://www.pecs-usa.com/

[17] http://www.mayer-johnson.com/ProdDesc.aspx?SKU=M125

[18] The State of Agile Development, 2008, http://www.versionone.com/pdf/3rdAnnualStateOfAgile_FullDataReport.pdf [on-line]

[19] Perez J. J., Guckenheimer S. *Software Engineering with Microsoft Visual Studio Team Systém,* Addison-Wesley Professional, 2006, ISBN 0321278720