Process of educational software development

Andrej Petráš, *petras@edi.fmph.uniba.sk*
Dept of Informatics Education, Comenius University Bratislava, Slovakia

Abstract

This paper describes iterative development of educational software, named Cube, for mathematics in elementary school. The aim of the program is to develop students' spatial imagination. Based on our previous experiments, we created this project, in which we tried to verify our skills from educational software's' development and define more accurately the whole process.

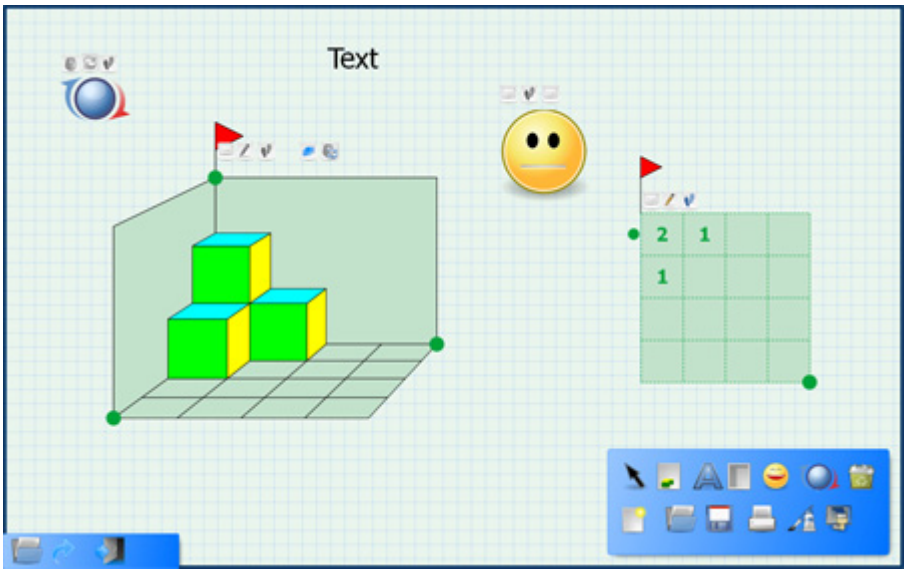

Figure 1. Screenshot of program's environment (http://cube.easyedu.eu)

It is a microworld for students, which is made for mathematics. Microworld gives children and students opportunities to build their own knowledge of fundamental concepts.

Many of our knowledge are achieved by co-operation. Students learn better when they can observe work of other people in their surroundings. The best conditions for learning are created when students are exposed to requirements, which extend their knowledge. Students like internet and ability of communication, changing of information and presenting themselves on the internet.

Keywords

computer literacy, educational software, action research, methodology of software development, experiment, project

Introduction

Our research's aim is the development of educational software. We would like to find out what methodology and methods are needed to use in order to obtain quality, good worked, but created in a short time and the cheapest software product and how this kind of development and software shape the cognitive process. Life in modern informative society and effective applying of the knowledge economy (see UNESCO) require modern education to develop new competencies in students, like critical perception and thinking, flexibly decision, controlling of unexpected situations, effective communication and cooperation. Informative and communicative technologies are important in development of these skills as resource. Educational software and internet put expression tool into another dimension. New technologies allow children to create, work or play. It is needed to think about questions: How do the proposal and development of educational software affect effectiveness of cognitive process? How affect methodologies of software's development and pedagogical research each other to achieve effective process of educational software's development?

We applied our current skills and knowledge of educational software's development in project Cube. It is microworld for students, which is aimed to mathematics. Microworld gives children and students opportunities to build their own knowledge of fundamental concepts. According to Kalaš and Winzer (2006), when students work with microworlds, they are encouraged to encounter concepts and relations through active engagement and exploratory learning. Next chapter deals with our development in theoretical aspect.

The Development of Educational Software

After evaluating and analyzing development of previous educational software, we came to the conclusion that one of the possibilities how to develop educational software is combination of agile software methodology with action research. We will analyze both these fields of study.

During the development of some products, it gets to paradox that customer is satisfied with product, which is relatively not usable. It can be caused by fact that when the customer requires some project, he does not know exactly all the properties of product himself. This is also one of the problems of educational software's development for children. If we want to define properties of certain educational software for children we need research, therefore we try to combine this research with development of software.

Agile Software Methodologies solve this problem by connecting a customer (pupil and teacher) with a developmental team. The term Agile Software Methodology denotes a group of methodologies, which assume that the only way how to check the correctness of the proposed system, is to develop it as quickly as possible, deliver it to the customer and use the feedback for improvements, see Kadlec (2005). Our development of educational software which we have already done is closest to this methodology. One of the groups of agile methodologies, which we use, is Extreme Programming.

Extreme Programming is effective, easy, flexible and funny methodology, which allows adding new versions of program continuously. It saves time spent by creating of documents and specifications. From the view of the Extreme Programming there are four basic activities: testing, code's writing, design and customer (communication with customer).

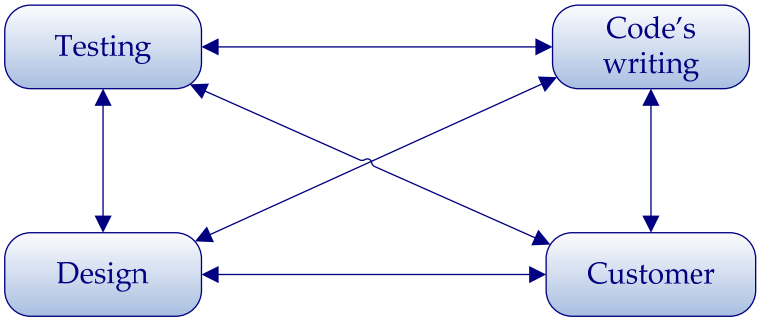

Figure 2. Interaction between elements within extreme programming, Kadlec 2004

According to Kadlec (2004), one of the most important elements of developmental team is expert, who rectifies development on the basis of his own research with children and according to this research he tries to define properties of software. By the term ***expert***, we mean a person from the academic environment who deals with media for children. His/her experience and knowledge should come forward throughout the development of the whole application. The expert should have a combination of academic knowledge and experience with production of pedagogic software. By *media for children*, we mean media for print, TV, multimedia, Internet or other technology.

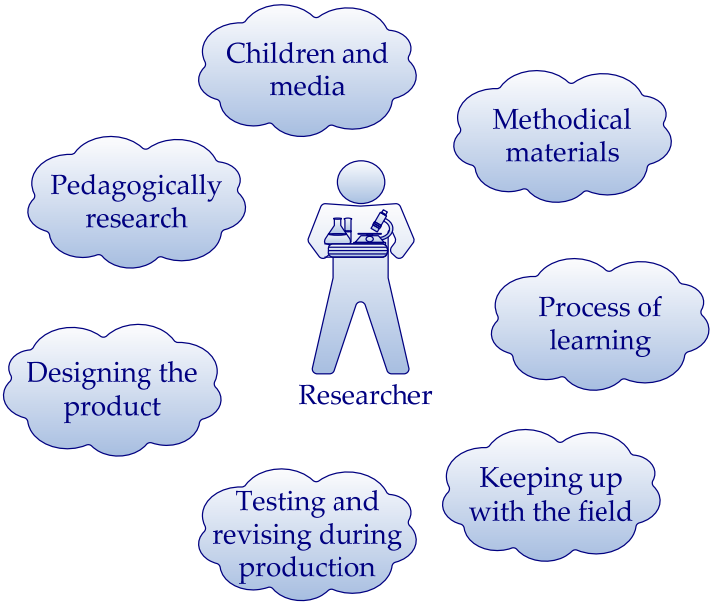

Figure 3. The researcher's role

In our project expert applies action research. There is a search for ways how to implement the results of research in practice and how to speed up the process of necessary changes. Action Research developed based on the critique of the traditional research, see Hendl (2005).

Action research is based on theory that effect of research is bigger, if all the people (teachers, students, employees …), who are connected with to research, will work together on it, see Hendl (2005).

Figure 4. Action Research

The course of the Action Research follows the terrain conditions and mostly uses the methods of qualitative research. Action Research needs to flexibly react to situations and obstacles and develop at the same time. Its cycles are the reactions to new knowledge and terrain problems. New cycles help to test and advance the interpretations from previous cycles, see Hendl (2005). Action research is applied research whose properties and progress are the similar to progress of agile methodology.

In our project we determined plan of software development, we put together group of students for this project and realized our program.

Planning iterations

Our plan was to create software, which will develop students' spatial imagination. We can divide this plan into two parts:

- In the first part, we studied some materials and realized two iterations with a teacher. Based on these studies we created our first design of environment and its functionality. We put together some students and we familiarized them with project.
- In the second part, children became testers and developers of software and we realized three iterations. During these iterations we applied to software some of students' and teacher's remarks, results of our research and we corrected some software bugs.

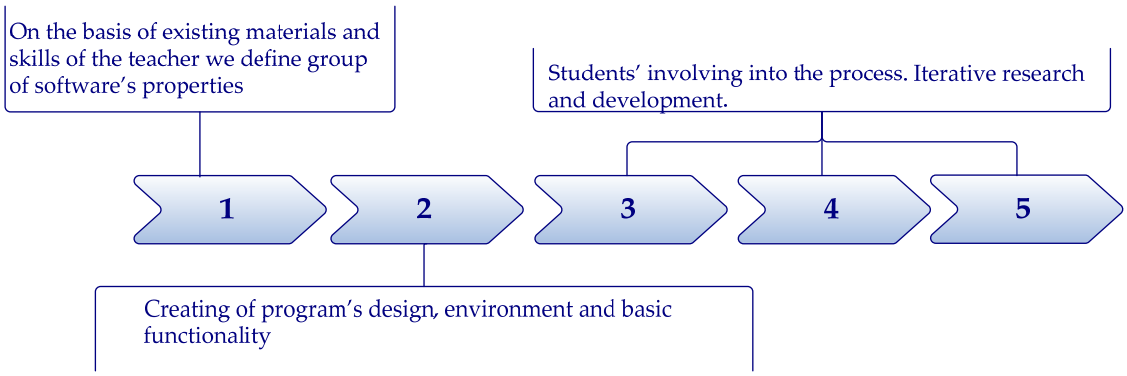

Figure 5. Planning iterations

Group of children

We involved students of fifth grade (aged 10 - 11 years) of elementary school Košická in our project. There are children, who feel like working with computer and they use it in their everyday life in the school or at home. Furthermore, they have computer training once a week at school.

In our students' group there were two boys and two girls. During the first iteration with students we had discussion about their using of computer, how and when they use computers and which of the programs they are able to use. Each the students wrote their own list of programs and activities.

We can divide students' answers into several groups:

- Programs – we involved these programs there: Imagine, LogoMotion, Anvil studio, Paint, Word, Power Point a others. Students mostly knew these programs from school and they also used them for doing their homework.
- Games – each of these students could list several games. In many cases, these games were aimed to force.
- Internet – number of answers and activities, which were written by students, was the same as those in first two groups. Many of different interacted games and activities were mentioned there. Students often search some information for their homework on the internet and they also like to communicate with other students by chats. Each of them has at least one email address and one student had his own "blog".

It is evident from discussion that students like, except games and homework, internet and ability of communication, changing of information and presenting of themselves on the internet.

Figure 6. Photo of students' group (Andrej, Jakub, Natália a Natália)

Project Cube

We analyse individual iterations of software's development.

Iteration 1 – In this iteration we had some meetings with teacher, who defined problems and her own experiences with current topic. She also proposed first software's properties based on her own experiences. She familiarized us with available schoolbooks, which are used by students.

Iteration 2 – After discussion with teacher we studied literature and we designed first environment and basic functionality. First design was draw on work books for students of elementary school. Then we began to program first version of our program.

Iteration 3 – After programming of first program's version, we involved group of students into project. Application was without sounds and it did not support editing of activities. It contained

only one task, but it was sufficient for students to create their own ideas about program's environment.

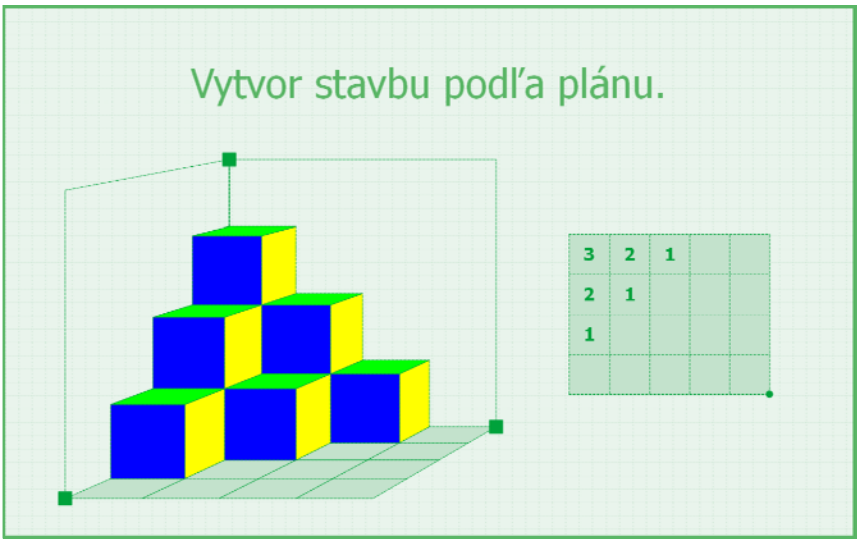

Figure 7. Screenshot of the first version

Students liked our program. We gave them papers, where they could draw their own imaginations about pictures and figures in program. Then we discussed about what and how program should do. There are some of students' notes:

- I should have possibility to choose, if I am girl or boy and according to this there would be different environment's colour.
- Program should encourage me, if I solve some task.
- I would like to make my own exercises.
- Cubes should have different colours and we could create different pictures by them.
- There could be also another shapes or models except cubes (it does not matter, that the name of program is Cube)

Some of graphical designs and designs of functionality were the same as our, but students also created unique designs, which we decide to involve into software's development.

Iteration 4 – We created version, which already contained objects for creating of activities. We could also save and load tasks in program. Some of new objects were created according to students' design. Students were able to work with program's activities quickly. After they did eight prepared tasks, they began to create their own tasks. They prove us that our tasks were easy and it was not problem to do them. Students changed and showed their own tasks among themselves. In this iteration we came to the conclusion that program would have more possibilities to be useful, if we created network version.

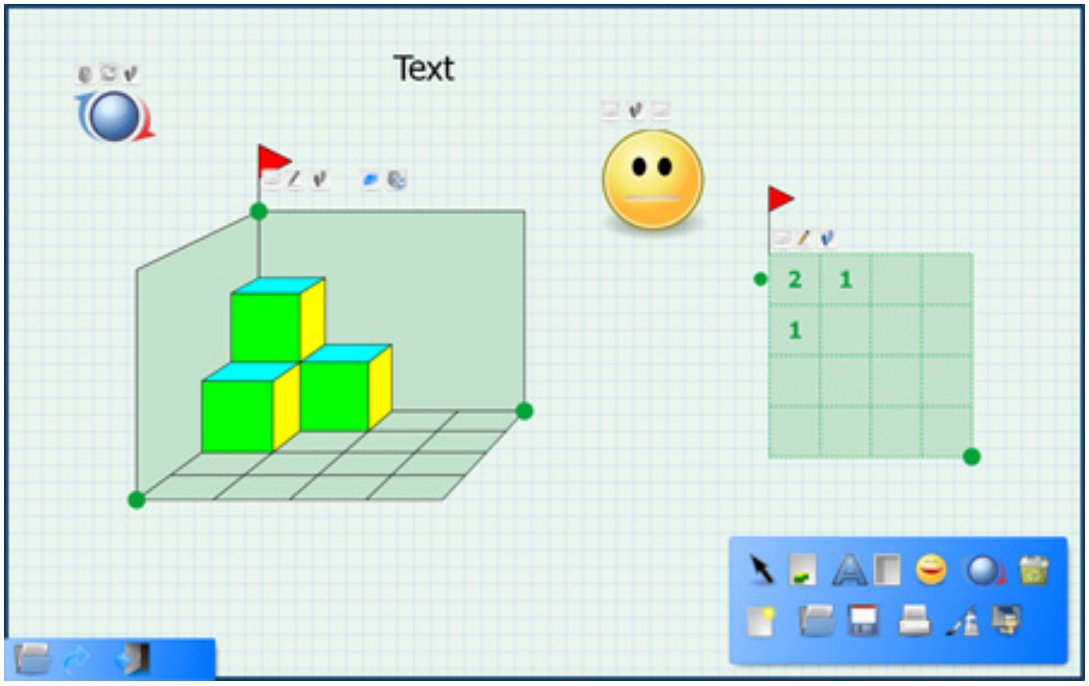

Figure 8. Screenshot of second version

According to Fisher (2004), many of our knowledge are achieved by co-operation. Students learn better when they can observe work of other people in their surroundings. The best conditions for learning are created when students are exposed to requirements, which extend their knowledge.

Iteration 5 - During the previous iteration, students always run our program on full screen. We observed that they do not like this program's property. Therefore, we decided to try to change program's environment, displaying of icons and to not run it on full screen, but in the window. There are several students' perceptions on this environment:

- Now, I can minimize it and browse the web pages.
- There are much more icons, but it is good.
- Previous environment was nicer.

Program does not contain more icons as previous environment and it missed some functionality, but students considered this environment more complicated. During this iteration, students found another program's bugs, which they did not find in previous iterations.

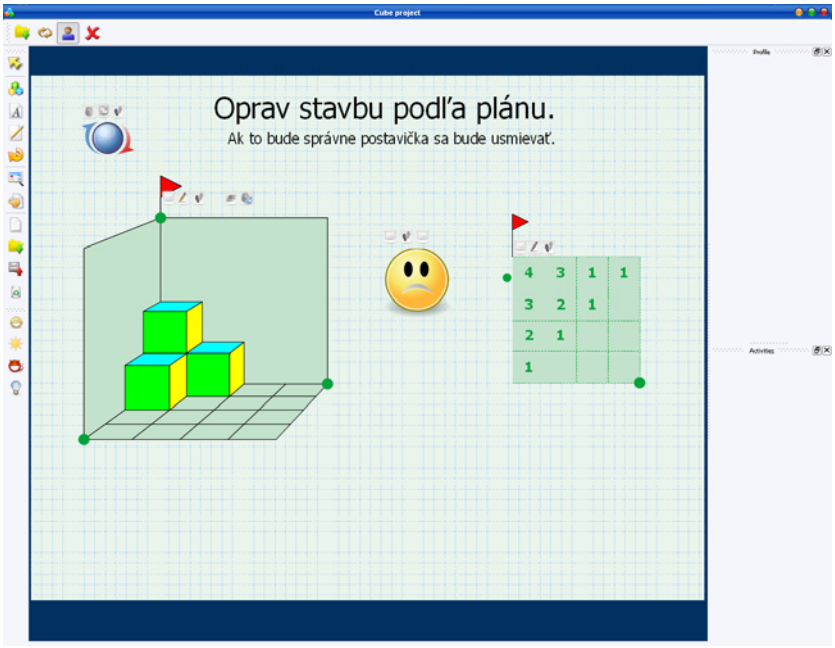

Figure 9. Screenshot of the third version

Process of software's development

Based on our experience in commercial projects and educational software developed so far, we experimentally tried the educational software development cycle based on Extreme Methodology and Action Research. The presence of the whole team at the lessons had a complement effect. Enthusiasm of children working with the program also supported internal motivation of all team members.

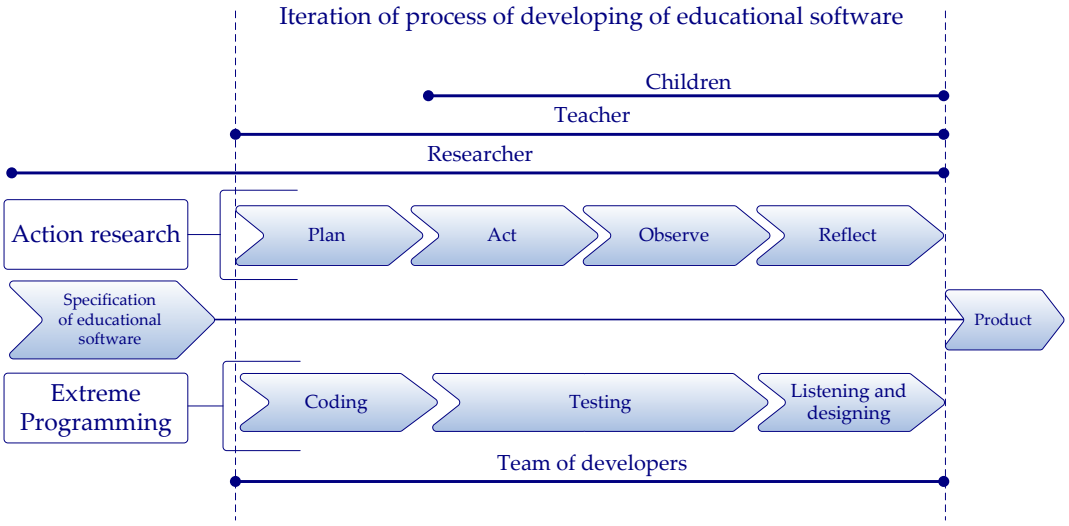

Figure 10. Process of developing of educational software inside project.

There were three people in the team: graphic designer, teacher and programmer. Only the teacher was present at the first testing iterations at lessons. Even though he was able to provide information about methodology and software errors, these lessons turned out to be inefficient from the development process point of view and few misunderstandings arose. When the whole team participated at the testing, the development efficiency increased. After each testing, we

discussed the lesson as a team and the results influenced the following iterations of either software development or pedagogic research.

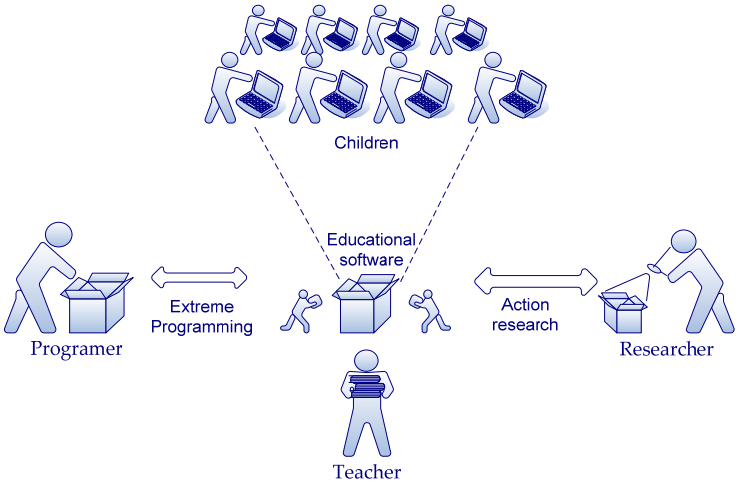

Figure 11. Abstract scheme of developing of educational software.

Conclusion

„What a child can do today in cooperation, tomorrow he will be able to do on his own “

Vygotskij, 1962

According to Unesco (2005), there are researches who prove that social co-operation can be helpful for students, when interactions are supported, which contribute to learning. Furthermore, interactions with other people are interested for students and they help to involve students into school work. Students intensively work because of results of their work. They know that it will be presented to other students by network.

Our experiments also proved that it is important for students to exchange information and findings among themselves. Therefore, it would also be supported by program, which will use Internet in it. Students will not solve tasks created by teacher but also tasks from other students. They will be able to create their own tasks to teach one another. In our subsequent research we will create network version and do additional iterations with students.

References

Kadlec, V (2004) *Agilní programovaní, Metodiky efektivního vývoje*, CP Books, Praha, ISBN 80-251-02364-8

Hendl, J (2005) *Kvalitatívní výzkum: základní metody a aplikace*, Portal, Praha, ISBN 80-7367-040-2

Fisher, R (2004) *Učíme deti myslet a učit se: praktický pruvodce strategie vyučovaní*, Portal, Praha, ISBN 80-7178-966-6

Unesco (2005) *Efektivní učení ve škole,* Portal, Praha, ISBN 80-7178-556-3

Druin, A (1999) *The Design of Children`s Technology*. Morgan Kaufmann Publishers, San Francisco CA, ISBN 1-55860-507-X

Kalas, I. and Winczer, M. (2006) *Building interfaces for on-line collaborative learning*. Journal Education and Information Technologies, Springer, ISSN 1360-2357, 371-384.

UNESCO (2002) Information and Communication Technologies in Teacher Education (A Planning Guide), Division for Higher Education