

The Concepts of Software Development for OOP and BAP Self-Learning Courses

Alena Zhmurko¹, Vladyslav Kruglyk¹, Michail Lvov¹

¹ Kherson State University

(alena.zhmurko@gmail.com, krugvs@gmail.com, lvov@ksu.ks.ua)

Abstract. The article describes the basic skills and phases necessary for the successful engineering and development of educational software. In the course of the analysis, the basics of software development are laid, including the objectives of the audience and the program, hardware and software limitations are determined, content resources are defined and auxiliary management tools are considered. Effective development of educational software for learning OOP (object-oriented programming) and BAP (basic algorithms and programming) self-learning requires the proper construction of the process.

Keywords. Self-learning, educational software, software development.

Key Terms. Academia, Development, KnowledgeTechnology, Research, TeachingProcess.

1 Introduction

A key technological issues of the information society in Ukraine is the computerization of social activities, including computerization of educational and research activities [9, 10]. Course management tools and educational software are different types of teaching and learning software. Blackboard, Moodle CMT, and WebCT – course management tools (CMT) which help faculty to manage and organize course resources for better assist in communicating with the students.

Those tools help to increase students' access to course materials. The primary goal of educational software, on the other hand, is to teach and assess students' knowledge. These products include many components.

Faculty of Information Technologies members frequently use administrative software and CMT with great success. Most IT courses and their associated university usually provide the hardware, software for supporting these programs [1].

The development of educational software requires a diverse skills that faculty rarely have the time to develop and can afford to support. To promote the creation of high-quality educational software, media development professionals such as teachers, aspirants and others specialists, multimedia artists, and software programmers should be key members of the project team.

2 Improvement from the First Steps

There are many advantages associated with the development process: it gives the team a single approach, facilitates budgeting and reporting on the work done, supports professional standards, increases the speed of development. A six-step software development process is described in the following sections of this article. This approach combines advanced teaching methods with practical aspects of software development. The ultimate goal is the development of innovative, educational programs.

Analysis. Analysis is the most important activity in the entire development process. At this stage, lays the foundation for the project. The better the planning, the better the quality of the final product. Throughout this chapter, carefully examine whether the PC is really the best means of delivering this information.

Identify the audience and goals of the project. This first step sounds very simple, but it is important. Developers should immediately answer the following questions and constantly contact them throughout the design and development process: what materials will this material use? What should the students know after the end of the lesson? In addition, indicate the basic knowledge of students.

Determine the limitations. Restrictions include the operating system of computers that will supply the program, the speed of the modem connection, the style requirements and the delivery method (cloud, USB devices, etc.). For web applications, select which browsers you want to use. In multimedia applications, video requires special. Be carefully planned.

Confirm the delivery. The educational software is more expensive than other forms of training, because of adaptation to the learner, the possibility of repeated repetition of instructions, self-evaluation with immediate and informative feedback, multimedia examples of authentic patients and easy access to training and other specific functionality. As a result delivery confirmation should verify that the instruction requires unique computer capabilities and cannot be accessed on another medium.

Information availability. Determine the availability of textbooks, other educational software, slides, source codes, video lectures and other teachers who can act as consultants and reviewers.

Development of management tools. To complete the project on time and within the budget and ensure its compliance with the objectives of the project, the project manager must develop a budget and a specification document. Depending on the project, specialized documents, such as a timeline or a content database, may be required. The software that helps to organize the process of developing educational software. He leads the project manager through the entire development process. It also includes several forms that help document and manage the project.

This type of tool give some opportunity: detailed time graphs for controlling development process. Moreover, it is good example for students or administrative workers of practice on developing the educational software.

Our software tool helps to create education process program within exercises and clear-presentation of the material.

Design. The design stage results in a cluster of documents known as the UI specifications that describe on paper all of the program elements and besides including listing of instructional content, and interactivity. Specifications frequently include a working prototype of the program to help enhance communication. Design guides, such as the Guidelines for the Design of Educational Software and Web/App Style Guide, can help ensure that all relevant design issues have been addressed.

Brainstorming technic. Brainstorming technic are usually include from three to five specialists. That is help to develop an advanced and creative design. During brainstorming, all ideas are welcomed, and no one is evaluated and not eliminated. The emphasis is on the number of ideas. Subsequently, all ideas are listed in a certain order and without any signs of belonging. Then everyone is discussed and either eliminated or saved. The revised list is reviewed again, except that the remaining ideas are listed in order of priority based on how they correspond to the objectives of the program. Some ideas will be discarded, based on the characteristics of the audience, its significance for the purpose, the constraint of the delivery system, the capabilities of the production team or the costs. During the review process, the teacher and the teacher will cooperate to determine the best learning strategy (textbook, drill, simulation, etc.).

Determine the appearance. Often the teacher's ideas about the "appearance" of the project differ from the ideas of the teacher.

To prevent misunderstandings, the appearance should be developed as early as possible and because of collective cooperation. Sketches can illustrate the designer's ideas or create a quick prototype using an authoring package, such as Moqups [3].

The navigation samples should be programmed to a level that the instructor understands how the program "works". The graphic elements must correspond to the delivery environment. Cloud storage has several limitations on graphics complexity, but the web server needs to consider the connection speed of the user to prevent a long download time.

Thus, for fast loading, you may need to trade complex and detailed (saturated) graphics. Mass media should be used sparingly and only to strengthen instructions. Animations are effective for attracting attention. Video is very effective for tasks such as demonstrating effects of interaction with the patient and dental procedures. However, the "talking head" video is rarely needed for training. Instead, you should replace it with a still photograph, accompanied by sound and / or text. When using video, try to keep short segments (usually twenty seconds or less) in order to keep students interested.

Development of a guide to the style of the project. The developer of training materials develops a document containing specifications for the presentation for the project [4]. The manual may include the use of a specific logo, font, color scheme, etc. It may

also include expectations such as scrolling symbols, writing style, image size, and the use of frames, video formats and screen size.

Consistency and detailed design. Together, the teaching staff (content expert) and the teacher conduct a very detailed analysis of the content. As a rule, the analysis of problems for procedural skills or analysis of the concept of principles and rules is completed.

Analysis of tasks requires a complex skill and breaks it down into component skills. Component skills are individually taught and then combined. This is often how the manual skills of dentistry are taught. First, the conceptual analysis reveals relevant and random characteristics of the concept, followed by a number of examples and improper examples. For example, a concept such as "diagnosing a sick mouth" can be taught using three characteristics (start, healing time and location) to distinguish between different diseases.

Block diagrams and storyboards. The designer of the training materials and the programmer use flowcharts to transfer the basic elements of the sequence of programs and interactions. The developer of educational materials creates a storyboard for visualizing the details [6]. Storyboards (see Figure 1 for an example) contain educational messages that students will see, including information presentations, questions, feedback, directions, charts and tips. Block diagrams and storyboards are developed simultaneously, because changing one requires modification of the other. This method is well suited for most educational programs, with the exception of simulations, games and hypermedia programs whose navigation routes are not linear and, therefore, cannot be easily planned using block diagrams and storyboards. In such cases, you can use the concept-mapping tool, such as Inspiration, to demonstrate the program structure.

Another option is to use the database to create a template with the dynamic information presented in the database field. Expertise. Content experts, in addition to those who work on the project team, should consider flow charts, storyboards, concept maps, databases, etc. For pedagogical quality, accuracy and consistency of content and suitability for the designated hardware and software. Reviewers should be free to ask questions and record comments. Corrections are made based on this information.

Development. Development is the process of projecting, programing, refining, and validating the program. The main steps are described below.

Write the educational program within lecture material, distinguish basic themes and practical examples for it.

Create graphics and animations. Animations should presentate work of algorithmic tasks. It is because simplification of educational program' understanding.

There are also a number of special purpose programs such as MatLab (eg. Mathcad, Simulink, Mathematica, LabVIEW, Maple) for creating basic environment.

Code the program. The specifications are translated into a language the computer can understand and that can be displayed as the instructional lesson. Approaches include using an authoring system, such as MatLab, a programming language such as C#, Python or Java [2], or an application that serves as an interface between the author and the computer code like CMS, etc. Currently, interactive programming is more difficult for cloud development than for web-application, but new software tools

are changing that. As individual parts of the program, such as graphics and media, are completed, they are added to the program. When all parts are assembled and all interactivity is turned on, you have the first version of the program. Throughout the development process, numerous changes and improvements will be made to the program. The project manager and the programmer will monitor a large number of changes using version control or versioning. This is a very complex process, especially when several people make a difference. For large programs, software for version control is important. For small projects, the project team must have an agreed protocol for renaming and dating files in each version of the program. In one document all assets and files, their contents and their changes must be registered. Programmers should insert comments explaining their code throughout the program. These comments, also known as documentation, should provide enough information so that the other programmer does not have to spend much time figuring out what you did. All developers should provide that someone else would have to make changes to the program. This documentation will save time and money for future programmers. Remember: documentation is more important than code.

Prepare auxiliary materials. The specifications will indicate which documentation will accompany the final draft. Major programs will have documentation for the student, instructor and / or technical staff.

Alpha testing. There are two levels of software testing. The project development team conducts alpha testing, and the evaluation expert, working with students, conducts beta testing. Both are important steps that should be taken with great planning and attention to detail. During alpha testing, the development team carefully checks the program for compliance with specifications, including appearance, design rules and functionality. The emphasis should be on trying an unexpected, not just what is stated in the specification. The team must consciously try to "break" the program. Based on the results, revisions are made, and these audits are checked. This process continues until errors are found.

Beta testing is often called a formative assessment. This is when the student uses the program, so that problems of design or programming can be recognized and solved. Although beta testing is described here as a separate process, it is not unusual for it to be intertwined with later stages of alpha testing. Carefully choose beta testers to represent a typical user, an advanced user and a beginner. Usually five to six students are enough. Ask each student to fill out the program, as if they are taking it out for credit and making their thoughts aloud as they move through the program. An observer (for example, the project manager, teacher or teacher) or videotape should record these thoughts. In addition, if a student does something unexpected, pay attention to it and ask about the action after that. After the student completes the program, the observer should discuss the user's comments and notes that they accepted.

At all times, the point of view of the student needs to be received [7]. An important result of the session is that you get the student's opinion about the structure and flow of the program, the clarity of the directions, the level of student control and other operational functions. Once again made changes. If major revisions are made, another round of beta testing is conducted; stop beta testing when only minor improvements are made.

Implementation and support. Implementation means ensuring the perfect, convenient and safe access to the curriculum by a student and a teacher. Implementation planning should take place at an early stage of the analysis. For a project, especially innovative, there is nothing unusual in using software and hardware that currently may not be available in the organization. However, this should not distract you from the implementation of an educational technology project. Instead, start working with the technology support staff at an early stage. Good communication can help alleviate the obstacles to implementation that you will eventually encounter, for example, access for students for new equipment.

Maintenance is an ongoing process of checking the relevance of content and software. After implementation, the training developer or programmer usually performs maintenance, monitoring, administration, and updating. To perform these tasks, they will need the hard documentation that you wrote. Finally, there is always a flurry of activity at the end of the project, during which routine tasks can be canceled. After the project is implemented, the project manager must complete any cleaning documentation, archival assets (images, video, etc.) and a file for copyrights.

Summative Evaluation. The total score determines the extent to which the objectives of the program are met. In the field of education, these goals may include improving learning outcomes, such as increasing long-term retention; Economics, for example, cost reduction; Or efficiency, such as faster receipt of content.

The first stage of the final evaluation takes place at the analysis stage, when the objectives of the project are determined. In conjunction with the assessment expert, the teacher determines how to measure the success of achieving these goals. It may also be advisable to carry out research or research.

Care must be taken to separate the influence of the delivery environment, in this case, of training technology, from teaching methods. For example, comparing a new educational program with a traditional lecture course entails not only changing the delivery environment, but also the teaching method. Many decades of research in the field of education have not yet led to substantial evidence that only one medium increases the level of education. However, the delivery environment can influence the economy, logistics and cognitive efficiency of training. Summative evaluation studies, with a quantitative evaluation of these results, provide valuable support for educational software.

3 Case Study: OOP and BAP Educational Process' Simulations

We described the approach to the development of educational software, but in fact, no project is like another project. Although it is important to have a process, there is no substitute for experience that will help to cope with these differences. The following research takes the described development process and overlays it with a healthy dose of reality.

Project Team, Analysis, and Design. The core team for Planning in educational process of two subject matter experts (for this project they are also the book authors

and so will be referred to here as the authors), the project manager, an instructional developer, aspirants (who also assisted with the book), and a narrator. The project is in progress, due to other educational tasks demands.

The goal of the article was to describe process of developing high-quality educational software.

Development. The process of collecting new information to improve the developed educational software continues.

There was finalized first version of look and feel, completed the charts, and refined the multimedia workbook and demonstration of main algorithmic tasks solves. The script for the introduction video and instructions audio are on developing.

Throughout the development process, tradeoffs were made that affected cost and program quality. The strategy used to code this program was based on preserving quality at the lowest cost.

Implementation and support. This program is designed for students to self-learning process through the case in front of a large group class or seminar, at which the case was discussed. Visual information (algorithms, examples, theory, algorithmic problems) from this case can help in the process of self-learning.

Since this project was to be included in the textbook, implementation issues were addressed at the analysis stage. At an early stage, it was found that the program should work on the network (cloud service). This eliminated any distribution problems. The code was documented as the program was developed and archived on cloud storages, etc. The modeling will be used in the planning process of the educational process over the next few years. At this time, it is expected that the product will be evaluated.

Comments and Lessons Learned. If the project team had been able to concentrate on the Cloud tasks resolves. The period resulted in personnel turnover. It also meant that multiple people fulfilled one role and pure progress are lost sometimes. For example, there was more than one manager developer. This turnover and role duplication reinforces the need for good project documentation. Good documentation decreases the amount of time required for a new person to understand the work that had been completed and shortens the learning curve, thereby reducing costs.

4 Conclusions

The potential of educational programs to improve the educational process for students is no longer a matter of dispute. The software provides graphical representations that allow students to focus on concepts and model problems that they could not learn before.

For programmers, software development has progressed to such an extent that it is relatively easy to write a program. This is the content of the program and the presentation format, which require most of the development time. Educational software is a growing market. As students and schools buy more computers, the demand for software will continue to grow. Improvements in the software, such as the Windows environment and other hardware, will continue to affect the types of programs developed.

The development of high-quality educational software is a multiphase process that requires a wide range of skills [8]. In this article, we were attempt to briefly describe the six steps in the development of educational software. That is why, in this article, we were focused on organizational and technical-organizational aspects. Careful analysis and planning are important to the success of the project. Time must be invested at an early stage in a complete and thorough design. Too much rush will inevitably lead to a less effective product or wasted during development. Development includes a number of activities that all must be carried out in a coordinated manner. Planning for implementation, maintenance, and final evaluation should occur early in the development process, rather than wait for completion. The above is should not be forgotten. The combination of all these phases will increase the likelihood of high quality of the project.

Many teachers can learn enough about software development for creating simple educational programs. Nevertheless, the wide range of knowledge necessary to fulfill the development tasks described in this article makes it difficult to create effective educational software that fully utilizes modern technologies. Again, we emphasize the quality of teaching. Of course, a small team has experience in several areas, while a large team of teachers, course designers, multimedia artists, programmers and evaluators will naturally have more experience. All teams of developers of educational programs, regardless of their size and qualifications, must recognize their shortcomings, and to develop a better product, consider the possibility of outsourcing work in which they are least qualified. The practical result should always be the same: do what is necessary to create the highest quality academic experience for students.

References

1. Lvov M., Kruglyk V.: Training of Algorithmization and Programming Using Python, (2014)
2. Kruglyk V., Lvov M.: Choosing the First Educational Programming Language, 188-198, (2012)
3. Higham L. S. Object-Oriented Programming: A Method for Pricing Options, (2016)
4. Goodrich M. T., Tamassia R., Goldwasser M. H.: Data structures and algorithms in Python, (2016)
5. Hilpisch Y.: Derivatives Analytics with Python: Data Analysis, Models, Simulation, Calibration and Hedging, John Wiley & Sons, (2015)
6. Spivakovsky O.: The Concept of Information Science Teaching at School and in Teachers' Training Higher Education Establishment, 21-25, (2003)
7. Oliynyk A.: Methodical instructions to implement practical work on discipline "Fundamentals of program's engineering", (2016)
8. Blake M.: A student-enacted simulation approach to software engineering education. IEEE Transactions on Education, 124–132, (2003)
9. Natsionalna doktryna rozvytku osvity, (2002)
10. Dodonov O. Derzhavna informacijna polityka i stanovlennja informacijnogho suspiljstva v Ukraini (2002).