CONTEMPORARY TEACHING PROBLEMS ON THE EXAMPLE OF AN IT TEAM PROJECT

Tomasz Szymczyk
Lublin University of Technology (POLAND)

Abstract
The article deals with the subject of learning by balancing on the border of self-learning and learning by doing, in relation to computer science and software development for ultra-modern technologies. It addresses problems of project management, risk management, time management, etc. in a two-semester IT project. Also described are the advantages and disadvantages of group work. Based on a conducted survey, students' most common ways of learning are discussed. The focus is especially on those areas of computer science where the availability of business books is very limited. This is due to the common practice of working with state-of-the-art devices and technologies such as zSpace, 3D scanners and Oculus Rift. In addition, the evolution of the team project theme over the last seventeen years has been described on the example of teaching in the Faculty of Computer Science at the Lublin University of Technology.

Keywords: higher education, virtual reality, teaching problems, learning by doing.

1 INTRODUCTION
The specificity of information science is its constant variability. Not only technologies, programming languages, frameworks, but also areas of IT application are changing. This entails the necessity of continuous training of both lecturers and students [2], [5], [6].

Continuous technological progress forces continuous learning to improve manufacturing processes. The learning process is the more difficult the greater is the complexity of the problem and the cost of the technology. Complexity means not only the range and level of required hard skills (e.g. programming language skills, expertise) and soft skills (e.g. time management, communicativeness, creativity, innovation), but also the need to know the context of the manufacturing process: what is the purpose of particular technologies, who will use them, etc. Lack of knowledge about the context often leads to theory parting with reality. In addition, the growing complexity of IT projects leads to an increase in risk factors which, when not properly managed (including: team management, risk management), can lead to confusion, decreased motivation and, in extreme cases, to panic and ultimately to failure. Teamwork therefore requires knowledge not only in the field of computer science, but also in management (people, schedule, resources, cost, quality, risk, etc.) and even the basics of psychology.

It is also necessary to have in-depth industry knowledge of the subject matter for which the project is being performed [11]. For example, when programming an application for a dental clinic, you need detailed knowledge of what the patient's medical documentation looks like, what types of procedures are being performed or planned to be implemented in the near future, how the body works, how patients sign up for doctors, treatments, etc.

This article describes how the aforementioned problems are reduced and eliminated during didactic classes conducted using the apparatus available in the Laboratory of Programming Intelligent Systems and 3D Computer Technologies [10].

2 METHODOLOGY
Regular teaching mode involves the acquisition of declarative knowledge, which can be applied in many cases and scenarios [7]. Often, however, the phase of implementation of developed theoretical solutions reveals a lack of the ability to use specific devices and a lack of understanding of the limitations of these technologies. Some of the technologies in the Laboratory of Programming Intelligent Systems and 3D Computer Technologies (LAB 3D) are provided to students as a tool for implementing programming projects. Students have the opportunity to try and use (hands-on learning)
often quite expensive equipment [12], [13], [14]. This limits the barrier to the availability of technology, often resulting in abandoning projects even before they begin.

The tasks for students include: getting to know the device specifications, testing their capabilities in a lab environment, testing and running experiments using their own programs and algorithms. The transfer of IT problems from the software space to the material space and work with industrial technologies enables knowledge acquisition in practice and allows for a better understanding of the possibilities and limitations of existing and discovered solutions. In this way, it is possible to create procedural knowledge [7].

The use of additional interfaces demands from students to move beyond the sphere of theory and learning by doing through the implementation of their solutions [8]. Additionally, participation in group projects it requires familiarity with many concepts and possessing many abilities. This means that they need to split tasks according to their skills. In this way, learning (including learning [9]) becomes an adventure (adventure learning), during which students not only learn about unknown technologies without the intervention of the teacher [8], but also learn their own limitations and abilities.

3 THE TEAM PROGRAMMING PROJECT - CHARACTERISTICS OF THE SUBJECT

The team programming project is a two-semester module of 30 hours and 3 ECTS per each part. Due to its level of sophistication and difficulty, it is often adapted to the general trends and needs of the labour market [1]. The subject of this article is taught at the master's degree level. It is a module that appears in the grid of classes in each of the specialisations in the field of computer science. This subject fully utilises the knowledge gained by students throughout the course of their studies. The topic of the project is most often suggested by the lecturer and is based on current trends, demand from the university or industry, or the available technical background. It can also be proposed after discussion with students. The first semester is focused mainly on the final project design. It uses UML in it. Students define the scope of the project, functional and non-functional requirements. Then they develop business project scenarios, and in UML – diagrams showing the static and dynamic aspects of the project. In this semester an application prototype is also implemented. The whole ends with the presentation of the project at the student public forum. The second semester includes: setting up a detailed work schedule with a breakdown into specific individuals performing them. A well-designed schedule is automatically a quick check for the trainer or the client whether the participants understand the project requirements well. The second semester ends with a public business and technological presentation of the running application. Then there is a general discussion involving the moderator and other students from the "competitive" groups. Presentations are designed to develop soft skills [3], [4].

4 EVOLUTION OF PROJECT THEMES OVER THE YEARS

Over seventeen years ago, the main burden of developing applications was a well-designed and optimised relational database. It was typical IT systems that used a database (MS SQL, ORACLE, Postgress, MySQL, Progress) and later chose information primarily from business. The topic varied widely in issues related to all kinds of "rentals". This was due to the clarity and comprehensibility of the ERD structure by students. These were very substantive projects, requiring a detailed overview of the functioning of such institutions. The following years saw the change of subject to servicing hotels, spas, hairdressing salons, ski slopes, tennis courts, etc. The technologies used by the students were ASP, PHP, HTML, CSS later ASP.net Visual Basic, C#. The technology was tailored to work optimally with database technology.

The following years brought quite different challenges. It was applications that recognised images. The subject "image recognition" or "advanced computer graphics" was not taught at any semester of study. Students had the task of developing their own library or getting acquainted with the possibilities of graphic libraries (mainly Emgu CV or Open CV). Mastering a ready-made library in the mid-2000s was difficult, mainly due to the lack of tutorials and manuals. Also worth mentioning is the language barrier: not all students had enough English. Very poor access to documentation made students write their own development solutions (libraries) or functions supporting graphics filters. They understood the essence of image recognition and with the help of the tutor were able themselves, using the optical method, to create projects such as the recognition of vacant spaces in an extensive car park. These
works were on the verge of independent experimentation and programming. There were some interesting security works: detecting moving objects, counting objects, estimating their speed, etc.

Lublin is a very crowded city. It was natural to elaborate on themes related to the creation of the so-called "green wave". Using parallel programming, student projects involved motion simulations at the largest, most sensitive Lublin junctions. Student design groups programmed the intersections and communication nodes and later tried to combine them by simulating a fragment of the urban agglomeration. Students encountered a completely new problem, not taught anywhere in computer science, which was the problem of communication in the traffic jam. These were also projects that directly mapped existing, actual intersections. It was not fiction but real simulation.

With the development of GIS technology, Google Maps and the general availability of GPS in smartphones there have been many projects referring to geolocation and surveillance. There are interesting applications monitoring courier work, supervising the safety of the elderly or children. Applications not only showed the current position on the map, but also analysed the data from the accelerometer.

Then the subject returned to the problem of image recognition. But this time it was the recognition of hand gestures. This was to create a series of non-contact interfaces for physicians, physical workers, and people with various conditions that prevent them from operating a computer mouse or touchpad. Attempts to create a visual interface were also made.

With the emergence of so-called virtual reality, the theme of the project has changed according to prevailing trends. The subject covered virtual projection on AR tags. Students first downloaded ready-made virtual, free, 3D elements, and in later years designed them themselves in Blender or 3D Builder. Applications presented, for example, various museum artifacts that were interactive with the AR tag. The tag motion caused the virtual object to move. These applications were also combined with GPS navigation and students created interactive guides for the most interesting places or objects. One of the most advanced was the virtual tour guide project at the Zoo in Zamość.

Students had hardware and software available all the time at home, but the quality of their work, their functionality began to fall sharply. It was necessary to "revive" the project theme. Fortunately, this coincided with the creation of the "3D Lab". Students received software, including Oculus Rift DK2 virtual reality glasses, interactive 3D zSpace monitor, Leap Motion non-contact manipulator, Phantom 3 PRO drones, and several Parrots. Missing items could also be printed on 3D printers: Makerbot Z18 and the very fine DWS x20. In the case of creating virtual worlds, students had access to scans made with professional 3D scanners: Artec Eva, Artec Spider and Faro Focus x33.0

Some have also opted for the Lego Mindstorms EV3 software, making them into interesting autonomous driving and broadcasting robots. Virtual reality has stimulated the creativity of students. Interesting projects have been created, such as the reconstruction of museums in VR, touch-screen operating systems, robots, maps, etc. Based on expanded reality, interactive storytelling facilities for children were created, etc.

Fig. 1. A look at the change in the issues of the Team Project over the years
The average yearly grade was calculated on the basis of all grades received by the students in the team project in the first semester (UML design) and the second semester (software in the selected programming language). The average consists primarily of the final evaluation of the project (functionality, aesthetics, responsiveness, technology selection, innovation) and evaluation for presentation of the project (presentation manner). The grades obtained in the software development process were also (if to a lesser extent) considered, not least for their consistency with the schedule. Such a definition of assessment reflects a in a clear and real way the involvement of students in study.

5 PROBLEMS OBSERVED IN TEACHING AN INFORMATION TECHNOLOGY TEAM PROJECT

Modern software development techniques force teamwork on one project. On the one hand, it is possible to use specialised version management software (Git, Mercurial, Bit Bucket, Team Foundation Server), automated application testing (Selenium). On the other hand, group work can cause conflicts and difficulties

Observed problems:
1) in the division into multi-person project groups there appears the problem of cooperation,
2) problems with the choice of team manager,
3) problems in creating a schedule,
4) the problem of timely (in line with the schedule) project implementation,
5) the problem of lack of time,
6) fear of having to master a new programming language,
7) fear of using ultramodern technology, which students do not have access to at home,
8) the problem of project scope limitation / project simplification,
9) projects for projects’ sake and not for industry.

The vast majority of these problems are due to the lack of so-called soft skills. Students are afraid to take responsibility for the project, and the project managers are chosen in a completely random way. Leadership characteristics are rarely taken into account. During team work, there are many conflicts in groups. This mainly concerns the choice of software tasks. Each student tries to choose the easiest, least time-consuming tasks. Also, performing one task is often dependent on the completion of another. Lack of "risk management", lack of sufficient communication, makes students want to do their own thing without asking the others for help. Often the ambition of one student causes "downtime" in the whole group's manufacturing process. It is only after some time that they learn to cooperate.
For many years it was possible to observe a clear reduction in the substantial and technological level of the projects. This was due not only to the lack of time, but also no willingness to create interesting projects. Perhaps the theme of the project also deviated from the interests of students. Students did not apply themselves to the project and did not have enough time or motivation. High ratings were not motivating in any way. The reason for this was taking up paid work in IT companies. For such students, full-time studies became an unnecessary addition. The aim was no longer to acquire thorough knowledge or high marks promising a scholarship, but the completion of studies in the least time-consuming way (with the lowest marks promoting to the next semester.)

In the team project students had to face a number of tasks that they had never learned in their studies (new programming languages, presentations in public, etc.). Not only scheduling, but even decomposition of the project into elementary tasks and estimation of the time of their execution was a huge problem for them.

Such acquisition of soft skills, teamwork, cooperation and responsibility for project development have certainly contributed to a significant development of soft skills and competences. The project also draws much attention to the way it is presented. Project execution is one thing, presenting it is another. After completing and testing the application, after assembling the relevant documentation, the last item to be evaluated is the business presentation of the project. This is a real challenge for most students. Presenting the value of the application, curiosity of the audience, the choice of eloquent vocabulary for technical students is a difficult challenge (with some notable exceptions).

A good and effective way of presenting an application, an idea, the approach to the subject, the rationale and method is extremely important in today's professional world.

6 CONCLUSIONS

With the emergence of ever newer IT companies in Lublin and the region, a marked increase in the percentage of working students has been observed. It currently concerns 60 to 95% of all students in the 2nd degree programme. Working for a company, besides financial satisfaction, gives the opportunity of education and development in the specific direction chosen by the student. Unfortunately it can negatively affect the process of study. Students are clearly tired and do not have enough time to learn. They often skip lectures, laboratories or practical classes. There was a clear tendency to pass tests at the lowest possible mark. Students began to consider studies as a supplement to work. In a very clear way, this also reflected on the final assessment of their part in team projects and the quality and functionality of the projects themselves. In order to curb, intensify and partially reverse this very negative trend, the latest research laboratories have been made available to computer science students. Thanks to that, a scientific research cooperation was established between lecturers and students. The latter have been given the opportunity to get acquainted with the latest high-end devices and technologies. In a very clear way, it stimulated their desire to learn and created very interesting functional projects, definitely at the highest grade level. By introducing innovative (for Poland) self-learning methodologies mentioned in this article, combined with interesting research and development topics, it has been possible to restore a high level of student education.

REFERENCES


