Risks in Students’ Software Projects

Tero Ahtee¹ and Timo Poranen²

¹Department of Software Systems, Tampere University of Technology, P.O.Box 553, FI-33101 Tampere, Finland, tensu@cs.tut.fi
²Department of Computer Sciences, University of Tampere, Kanslerinrinne 1, FI-33014 University of Tampere, Finland, tp@cs.uta.fi

Abstract

Software project education is an important part of software engineering studies in all universities. A well written final report gives an overview how the project went and what students have learned from the project. In this paper we analyze risks in 76 final reports from students’ software projects from academic years 2006-7 and 2007-8. The projects were done in two Finnish universities. We recognized that four major risks are tools and skills to use the tools (61% of projects), technological problems (53%), scheduling problems (61%) and working or studying too many other courses during the project (45%). We also give guidelines to teachers and project managers to take into account the special nature of student projects.

1. Introduction

Almost all universities where information technology is taught have a compulsory project course on software engineering [5,6]. In these courses, students work in project groups and construct a relatively large software product. Important learning goals in this kind of course include making students familiar with project work in general, but also to integrate the knowledge and skills that students have learned in various earlier courses. Students should also realize the purpose of documentation and testing.

Our project courses are compulsory and they are recommend to be taken on 3rd or 4th studying year. The goal is to design and implement a software product in a group having 5..8 team members including a project manager.

In this paper we study how project risks are analyzed in final reports at Software Engineering Project courses at two universities in Finland. We review our risk findings from 76 students’ software projects performed during academic years 2006-7 and 2007-8. Our main findings are that most often met risks are scheduling, development tools related problems, working and studying too many other courses during the project, and technological problems. Finally we summarize our findings and discuss on future research directions.

2. Finishing a student software project

Software Engineering project courses have been taught over 20 years at Tampere university of Technology, Department of Computer Sciences (UTA) and Tampere University of Technology, Department of Software Systems (TUT) [2]. Usually the project groups' working hours are in total something between 700 and 1700 hours. The project topics are real, and all teams produce a different product.

The course takes a whole academic year, it starts in the beginning of fall and end on spring. During the course students produce at least a feasibility study, project plan, requirements specification, design document, test plan, test report, code and final report. Depending on the topic, a team might produce other documents as well (e.g. user manual and maintenance
guide). All teams send weekly reports to the course lecturer and to the other stakeholders. During the project there are meetings with the course staff and inspections [1] for main documents and code to ensure good quality of the process and product. The teams can use any widely known software development model.

In Finland it is quite common that computer science students work during their studies. We estimate that more than 50% of students work at least part-time after receiving their B.Sc. degree. About 10% of students are working full-time.

The project closing process in software project courses [2] at TUT and UTA, includes product delivery to the client, writing a final report, a public project presentation, and a closing meeting with the course lecturer. The final report is discussed in details with the project team in the final meeting. The final meeting takes from one to two hours. The meeting is a very educative and according to student feedback, it is the best course activity from a learning point of view.

3. Final report analysis

During academic year 2006-7 and 2007-8 there was in total 79 students’ software projects at TUT and UTA. The courses are similar and they use the same document templates so it is reasonable to analyze their final reports together. Two projects produced a final report with so many missing details that they were removed from the analysis. One other project quitted so it didn’t produce a final report. Therefore we used in our analysis final reports from 76 projects.

In 17 projects the daily working language was English due to exchange students participating the courses. 43 projects had a company, 16 a university unit and 10 a non-profit association as a customer. 7 projects used students’ own topic and therefore they didn’t have a real client. The average amount of work done in the projects was 1217 hours. Topic was a www-application with a database in 33 projects, 28 did a stand-alone application for Windows or Linux, 11 were mobile phone applications and 4 embedded system implementation projects.

Common risks in industrial software projects have traditionally been personnel shortfalls, unrealistic schedules and budgets, and developing wrong functions and properties [3]. Boehm and Port [4] have also studied how risk management should be taught to students.

To find out which risks are common in student projects, we classified met risks (both foreseen and unforeseen) from the final reports into 14 categories. The categories were quite easy to recognize since many themes occurred regularly in the risk analysis of the projects. We didn’t analyze how low or high impact problems caused to the team. But since the teams reported these met risk items, their impact was at least low. In Table 1 we list the recognized risk item or risk category and how many projects reported the item. The four major risk categories are bolded. All projects reported at least one risk, and the average amount of met risks was 4.7. The highest amount of risks in one project was 9.

Communication problems between the group members and with the client were usual. 24 projects (32%) reported that they had difficulties in communication. Common problems were missing or slow replies from the client, and difficulties on communicating with some team member.

Illness, injuries and social problems caused problems to 26 teams (34%). There were some serious injuries too, but most of the problems were caused by a short-term flu or fever. Also a few social problems with the girl/boy friend were reported. In two projects a death of a relative caused difficulties for a team member to participate project work actively.
Table 1: Risks in students' projects.

<table>
<thead>
<tr>
<th>RISK NAME</th>
<th>NUMBER OF PROJECTS</th>
<th>PERCENTAGE ON HOW MANY PROJ.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Communication problems</td>
<td>24</td>
<td>32%</td>
</tr>
<tr>
<td>Requirements</td>
<td>24</td>
<td>32%</td>
</tr>
<tr>
<td>Illness and social problems</td>
<td>26</td>
<td>34%</td>
</tr>
<tr>
<td><strong>Tools and skills</strong></td>
<td><strong>46</strong></td>
<td><strong>61%</strong></td>
</tr>
<tr>
<td>Quitting team members</td>
<td>14</td>
<td>18%</td>
</tr>
<tr>
<td>Process problems</td>
<td>15</td>
<td>20%</td>
</tr>
<tr>
<td>Motivation level low</td>
<td>27</td>
<td>36%</td>
</tr>
<tr>
<td><strong>Technology problems</strong></td>
<td><strong>40</strong></td>
<td><strong>53%</strong></td>
</tr>
<tr>
<td>Documentation problems</td>
<td>9</td>
<td>12%</td>
</tr>
<tr>
<td><strong>Scheduling problems</strong></td>
<td><strong>47</strong></td>
<td><strong>61%</strong></td>
</tr>
<tr>
<td><strong>Working and studying during project</strong></td>
<td><strong>34</strong></td>
<td><strong>45%</strong></td>
</tr>
<tr>
<td>Client related problems</td>
<td>18</td>
<td>24%</td>
</tr>
<tr>
<td>Third-party components</td>
<td>15</td>
<td>20%</td>
</tr>
<tr>
<td>Group work related problems</td>
<td>14</td>
<td>18%</td>
</tr>
</tbody>
</table>

24 teams had problems with requirements gathering and understanding requirements. A common situation was changing requirements after freezing the requirements specification document. Also in some projects it was really hard to get client’s opinion for some detail.

Over half of the projects (46 projects, 61%) met difficulties on using development tools and programming languages. One reason may be that not all users can be productive after one or two-month use of a new tool. Development process related problems were reported by 15 projects (20%). Unstructured meetings, task division problems, development model related problems and management problems were counted to this category.

Technological problems, like breaking hard disks, motherboards and displays, non-working USB-sticks, printing and internet connection problems were seen in 40 projects. Also in some mobile phone projects technology changed too fast.

Poor documentation quality caused problems to nine projects. Quite often an important detail was missing from the document. Some teams forget to update their documents.

27 projects had at least some problems caused by low motivation. A team member quit in 14 projects out of 76. Unfortunately often the lost member didn’t tell that he/she will quit and the rest of the team tried to contact to the lost member long time before the quitting was realized. The motivation problems and loss of a member caused more work to other members, and therefore often scheduling problems which were most common from all risks. 47 (61%) projects were not able to follow their initial schedule, and they had to change the schedule unexpectedly. The amount of needed work for a single task or for the whole project is very hard to estimate in commercial projects, but it is even more difficult in student projects. Very often scheduling problems were compensated by reducing requirements.

In 34 (45%) projects working or studying too many other courses at the same time with the project were seen as a problem. Scheduling problems, and working and studying during the project seem to have a close connection.

Client caused problems in 18 projects (24%). To this category all client related problems, except communication, were taken account. Often the client was not able to give on time testing equipment or test data that was agreed in the project plan. In one project client decided to quit the project (students still continued the project without client to complete the course). In a few projects client didn’t gave enough feedback to the team. Naturally those projects which had their own topic, didn’t met any problems with the client.
Almost all projects that used third-party software components had difficulties with them. Interfaces didn’t function as was promised or documentation of the components was not understandable or sufficient.

Group work and working environment related problems were met by 14 teams. A common difficulty was that some team members didn’t get on together. Also problems on finding a common place to work and meet or deciding a common meeting time was counted to this category. Sometimes it had a lot of arguing about which tool to select.

Risks in student projects seem to be slightly different than in industrial projects. It is clear that there are no possibilities to run out of budget, because the projects do not deal with money. Student projects seem to have more difficulties with the tools and scheduling than industrial projects, even if the university projects are in general smaller than commercial projects. Also working at the same time with the project can be seen a major risk. To reduce the risk project manager should be aware of work situation of the team members, and take this into account in the project’s schedule. Other main risks can be prevented and mitigated by some sort of training.

4. Conclusions

We strongly suggest that every project course should have a written final report, and a final conclusive meeting with the project group that finalizes the learning process. Our analysis showed that the teachers and project work team members should pay a lot of attention to support and training on using development tools and project planning. Risk item list given in this work should be discussed with the course students and clients to make all stakeholders aware of common project risks.

Our risk analysis should be done more carefully in future. We didn’t take account severities of the risks, and we didn’t analyze correlations of the team’s working language, how many different nationalities were in the team, project’s topic, and client’s organization. It would also be interesting to recognize causalities between risk items. Different categories were quite expansive and they should be divided into more specific categories (for instance, requirements related risks can be divided into requirements gathering, requirements analysis and requirements change management problems).

5. References