Categorizing Projects for Software Engineering Capstone Courses

Attiqa Rehman [0000-0003-0375-9867] and Monica Divitini [0000-0003-2323-2237]

Center for Excellence in IT Education (Excited), NTNU, Trondheim, Norway
{attiqa.rehman; monica.divitini}@ntnu.no

Abstract. Courses centered around a capstone project are common in Computer Science (CS) and Software Engineering (SE) programs. They are normally offered towards the end of the study program and give students the possibility to work on realistic and engaging projects, at the same time giving them hands-on experience of working with external actors, hereafter customers. In this paper we investigate the type of projects that are proposed by external customers. The study is based on a specific course at our university, where capstone projects are proposed by external customers from industry, start-ups, non-governmental organizations, and research organizations. In the scope of this work, we analyzed the 134 project proposals that were submitted for our course over the last five years, 2018-2022.
The research question that we are addressing is: Which type of projects are proposed by external customers? To answer this question, we categorize the submitted project proposals into nine categories: compulsory or volunteer system, product driven or goal oriented, new product or enhancement, general audience or specific target group, information system or embedded system, sustainability aspects, type of license and competency-based classification. By classifying the proposals, we identify trends that might be used as a starting point to reflect on course and program organization. Based on our study, we suggest classifying projects from external customers as an important strategic tool for helping teachers to reflect on their work.

Keywords: Capstone Projects, Project Classification.

1 Introduction

Project-based learning (PJBL) is a pedagogical approach often used in Computer Science and in Software Engineering (SE) education to help students develop multiple skills in the context of a single course [1]. In the context of specific courses, the focus generally remains on the subject matter. For example, in a course on requirement engineering the focus will tend to be on requirement-related processes rather than e.g., on budgeting, planning or coding [2].

In order to focus on the overall project experience and bring together competencies in a more holistic way, courses based on capstone projects are often adopted. During ``*the capstone projects, students apply the previously learned skills, deepen the*

understanding of the acquired knowledge, extend the area of knowledge, and apply their knowledge and skills in a realistic simulation of professional experience in the university environment.`` [3], pg.1. Capstone projects are recommended by Association for Computer Machinery (ACM)/ Institute of Electrical and Electronics Engineers (IEEE) [4, 5] for both the Computer Science and Software Engineering undergraduate programs as well as by Software Engineering Body of Knowledge (SWEBOK) V3.0 [6]. Capstone projects in Software Engineering have long been a topic of research. The approach is reported as helping students to develop multiple skills as well as their learning capabilities [2, 7]. Capstone projects also play a critical role in bridging the gap between the skills developed during studies and industry demands, playing a critical role in the professional life of graduates [8–10].

A recent literature review pointed out the wide adoption of capstone projects, but also the different ways they might be implemented [11]. Capstone projects might vary in terms of e.g., credits, duration, number of students. One important difference is the involvement of external clients. According to the review (ibid.), only 58% of the courses included in the literature survey involved external clients. Main reasons for not involving external clients include lack of external contacts, time issues, and in general projects with clients are considered as more cumbersome for the teaching staff.

At the same time, the involvement of external clients, or customers in the terminology that we use, has known benefits. In general, pedagogical approaches with a strong focus on work are known to enhance the achievements of learning outcomes at the same time helping students to adjust better to work-life needs [12]. Projects with real customers are also reported to help achieve the learning outcomes for SE program and bridge the gap between academia and industry [10]. They also promote the development of key professional skills like knowing how to interact with a client [13].

In this paper, we focus on capstone projects that are proposed by external customers. We contribute to the research in this area by investigating the type of projects that industry is proposing. With this effort we aim at (i) identifying emerging trends and (ii) reflecting on the implications for course and program design. The study is based on the analysis of the 134 project proposals submitted to one of the capstone projects at our university during the period 2018-2022.

The rest of the paper is organized as follows: Section 2 provides the description of the capstone project course analyzed in this paper; Section 3 presents the research methodology; Section 4 the categorization of the projects; Section 5 a detailed analysis of results; Section 6 reflects on the implications of the results for course and program organization and finally Section 7 concludes the paper with some future directions.

2 Our Case

The capstone project course that we are analyzing is mandatory for students in the last semester of their Bachelor's degree in Informatics. The course is 15 ECTS (European Credit Transfer System). If we consider the classification of capstone projects presented in [11], our case can be classified as a course spanning over one semester; with a team size of 5-7 students; with external stakeholders proposing project ideas and acting as

customers; software prototypes and reports produced and assessed during the semester, but with project phases and technologies varying depending on the characteristics of each project. The goal of the course is to ensure that each group delivers a product accepted by the customers. Students develop applied skills in software engineering as well as specific skills about programming languages and tools depending on the specific project they work with.

A description of the course and its organization is beyond the scope of the paper, that focuses on the projects proposed by external customers. We therefore limit our description of the course to the initial phase of collecting the project proposals. External customers are contacted a few months before the beginning of the course through the personal network of the teaching staff and are invited to submit proposals. Customers are informed about the general constraints of the course in terms of time and workload. Apart from the general constraints, customers are free to propose any project that they consider relevant and feasible to complete within the given timeframe. The suitability of the proposals is then assessed by the teaching staff.

Customers are asked to use a predefined template to present their proposals and to stay within 1 page. Proposals are kept short to limit the workload for the customers. Short proposals also make it easy for students to go through all of them quickly at the beginning of the course and express their preferences. These proposals are the documents we analyze in this paper. The template is provided in Table 1.

Table 1. Project Proposal Template

1. Name of the company and contact information.
2. Language of the report, by default English
3. Project description. Customers must provide a brief description to help students select the project during the initial meeting. More information can be provided later. The project must have clear boundaries, to evaluate the group's contribution. We do not accept a generic task where students will work on the overall company product, contributing in a fragmented way to distinct parts. The project should involve design, development, and testing of the expected product. All projects should include a development phase, so we do not accept, e.g., tasks that only involve a literature review, product or market analysis, user evaluation of an existing product, ...
4. Suggested number of students, by default 5-7
5. Competencies, distinguishing between competencies required and the ones that are considered useful but not critical.
6. Software License under which the product will be released. By default, if not differently specified, we assume that the software produced will be released under an open-source license.
7. Any additional information that customers want to provide.
8. Plan B. This field was added during the pandemic to make sure that customers consider challenges, for example with accessing specific resources.

The template remained stable in the years, with (8) introduced in 2021 and (6) introduced in 2022 to clarify a project element that might cause conflicts later in the project.

3 Methodology

For this study we focus solely on the content of the project proposals with the aim of identifying the type of projects that external customers are interested in and the implications for academia. More specifically, we analyze the 1-page project proposals submitted using the template presented in the previous section in the period 2018-2022.

After collecting all the proposals from the course organizers, we categorized them combining an inductive/deductive approach [14]. Given the nature of the documents, our goal was to identify categories that could help us to clearly characterize the proposals and identify trends in the course project portfolio.

Based on the experience with the course and existing literature, the authors identified an initial set of categories as relevant themes to analyze the documents. The authors then read the documents, categorized them, and revised the categorization as needed. The process was highly iterative. The first iteration on a sub-set of proposals was conducted by the two authors to assure a common understanding of the categories. The first author then proceeded with the analysis of all the documents, in an iterative way and with revisions based on discussion with the second author. The first author does not have experience with the course. This helped us to look at the documents with a fresh eye, avoiding possible biases due to previous knowledge of the material. The conversation with the second author, with extensive course experience, added reliability by avoiding the risks connected with a single person analyzing the data.

The analysis was conducted in Microsoft Excel year-wise. At the end, the graphs presented in Section 5 were generated to provide an overview.

4 Capstone Projects Categorization

Capstone projects are categorized differently by different researchers. In [15] the authors analyzed the capstone projects of a SE graduate program, finding that from agriculture, business, education, hospitality and social, different domains appeared in capstone projects. In [16], capstone projects are divided into two classes based on the proponent, either industry or faculty. Further, capstone projects are categorized into sub-categories, namely process models used to develop; intendent audience; platforms for which they were developed; and the programing language used to develop. The used classification also considers project source code availability, e.g., the software is open source; and whether the project is started from scratch, or it is an enhancement of an existing project. In [17] the researchers categorized the domains of capstone projects, including network, wireless technologies, data mining, business intelligence and computer vision. The study in [17] also presents the different types of deliverables at the end of a capstone project, with most of the projects expecting the delivery of a working software prototype at the end. Some courses may also require additional or alternative forms of deliverables like design specification, requirement analysis, etc. A study presented in [18] highlights the importance of capstone projects in the field of Computer Science. The capstone projects are classified into service and production sectors where each of them is further classified into eight different categories. The service sector is

categorized into healthcare, hospitality, distribution center, call center, nursery home, transportation wholesale, and education. Whereas the production sector is categorized into high tech, machining, pharmaceutical, textile, plastic, agriculture, food, and construction. The researchers in [19] classified the software applications using a machine learning algorithm. The algorithm returned twenty-two different categories, including e.g., graphics, web, database, and bio-informatic. Sustainability in software projects is an important and hot research topic. In [20] authors stated that students must be able to work on existing code so enhancing the features of an existing software is equally important as that of creating a new software. In [21, 24] the authors highlight the importance of sustainability in customer-driven project courses, with focus on the UN Sustainable Development Goals.

As we can see from this brief overview, there have been different attempts to classify capstone projects. The categorization that we use in this paper is mainly based on existing project categorizations. In [24] the authors stated that software projects can be categorized as volunteer or compulsory system, product oriented or goal oriented, information system or embedded system, the software projects can be for a specific audience or for general audience. So, Items 1-5 are borrowed by [24], though they might appear in slightly different forms also in other categorizations. As stated in [22], the three pillars of sustainability are social, economic and environment. So, we analyzed the sustainability aspects present in the capstone project proposal and presented them as item 6 named sustainability. Items 7, type of license, is introduced based on our experience with an important aspect of projects with external customers. This type of categorization is also used in [16]. In [19] the authors provided the categorization of projects according to different domains like web, database, front end, bio informatic etc. During the analysis of project proposals, we categorize the projects according to the competencies required to develop the Item 8 emerged as result of this analysis. For each category, we are presenting below a short description with examples.

1. Volunteer or Compulsory: A project is categorized as compulsory if the produced system must be used to complete a task, e.g., a system to control the temperature of a building. If the system for temperature control is not used it may lead to overheating a building or not-functional heating system which leads to a problem of inefficient utilization of energy. A project is categorized as Volunteer when the users of the final prototype may decide to use it or not, e.g., a computer game-based language tutor.
2. Product Driven or Goal/Objective Oriented: At the end of a project the product needs to be delivered, or certain goals (which are stated as artifacts in [17]) are achieved. E.g., a project that requires students to develop a portal to promote cooperation with parents in kindergartens is classified as product driven. A project intended to improve the performance of a routing protocol is placed in the Goal oriented project category.
3. New Product or Enhancement: The product can be designed and developed from scratch, or it can be the enhancement of a previously delivered product e.g., adding new features into a virtual reality-based job portal. Understanding and making changes to existing code is a very important skill. So, for capstone projects the

competencies required for enhancements of existing products are equally important than those for creating new products [20].

4. General Audience or Specific Target Group: This category considers if the developed product is intended for anyone, then the project is classified as for general audience, or it may be intended for a specific target group. For example, if a project proposal states that the product is intended for people who are visually impaired, this means that this project is intended to a specific audience group. A virtual reality-based city tour where the audience group is not specified is classified for general audience.
5. Information System or Embedded System: The software used to access or provide information are information system. The database system, web-based system, system accessed via internet, service sector software etc., fall under the information system category [15, 17–19]. The software may use to control the machines in this case it will be considered as embedded system, e.g., a robotic arm is considered as an embedded system because in this project the robotic arm is controlled via software whereas a system getting daily inputs from parents of kids at kindergarten about their presence is considered as information system.
6. Sustainability Aspects: in this category, we consider three dimensions of sustainability, namely environmental, economic, and social[21–23]. In the environmental aspects, we consider how a software system is decreasing the carbon footprint by efficiently utilizing the natural resources. Economic aspects consider if a project can be used to decrease the costs in terms of money and its business model allows it to sustain itself. Impact on society is a category used when a system explicitly addresses its impact on society. For example, if a project is using software and hardware (sensors) to decrease the heating costs in government buildings, it is providing a solution that addresses aspects of environmental and economic sustainability. As the energy consumption of a building decreases, its carbon footprint also decreases. At the same time, utilizing less energy means saving money, so it is also considering the economical aspect of sustainability. If a project purpose is to improve the health and wellbeing of a group, like for example migrants, then we place this project as considering the impact on society.
7. License Type: This category considers under which license a product is expected to be delivered. This field is filled in by the customer in the proposal (Item 6 in Table 1). In some project proposals this field was missing, and they were placed into the not mentioned category. The authors in [16] also use the same category.
8. Competency-based Classification: This category is highly dependent on the project proposals provided for analysis. Each project proposal has a field named Competencies (Item 5 in Table 1), where the required competencies are listed e.g., Front-End development, Mobile Application Development, Sensor etc. So, just like [15, 17, 19], we analyzed the available data in terms of project proposals and categorize the project according to the competency required for its development. At a time, a project may appear in more than one category e.g., a project may have a web site as well as a mobile application. This category can take the following values:
    i. Web
    ii. IoT

iii. Mobile Application
iv. Client Server
v. Desktop Application

5 Results

In this section we present the findings concerning the categorization within the projects` proposals of customers.
For the analysis, we used a total of 134 proposals, distributed as following: 25 in 2018; 23 in 2019; 29 in 2020; 28 in 2021; 29 in 2022.
Figure 1 shows the number of projects proposal intended to develop a volunteer system or compulsory system over the last five years. Figure 2 shows how many projects are for a targeted audience and how many of them are for a general audience. Figure 3 shows the year wise projects producing a product at the end or achieving a specific goal. Figure 4 depicts how many of the projects are a new product or are enhancement of a previous product. Figure 5 shows the distribution projects as information system and embedded systems. Figure 6 shows the year wise projects license status. Figure 7 shows the sustainability aspects considered each year and Figure 8 depicts the classification of projects according to technology. Figure 9 and Figure 10 further elaborate the results by showing the accumulated results of five years as a pie chart in terms of sustainability and technology-based classification.

5.1 Trends

It was a hypothesis that sustainability, being a hot topic, would appear in the proposals submitted by customers. However, we found that the issue of sustainability is not addressed explicitly by many customers. As we can see in Figure 7 and Figure 9 on the overall, 75% of projects consider sustainability, but this ratio becomes lower as there are projects considering more than one aspect of sustainability simultaneously. We also do not see a significant increase in the number of projects addressing sustainability throughout the period, i.e., from 2018 to 2022. This seems to contrast with a significant increase in the awareness of issues connected to sustainability in academia, but also in industry, public sector, and among the general public. At the same time, the result of this study is fully in line with the results of a similar study presented in [21, 24]. It is therefore important to identify ways to make sustainability issues more visible and increase awareness among customers.
Figure 4 depicts that over the years 2018-2021 the number of new products is always greater than the enchantments in the previous projects. However, just after the covid the year 2022 shows an increase in the number of enhancement projects. The reason may be that the projects developed during the covid time need improvements, however, for this we need to research on customer satisfaction from the capstone projects developed by students. The technology classification shown in figure 8 concludes that the web-based projects remained in trends throughout the time from 2018-2022. The mobile application projects are second highest after web. IoT and client server are presenting about the same ratio, and Desktop Applications are lowest. From Figure 2 it is

possible to note that most of the systems are intended for a specific group i.e., developed for a targeted audience. These projects might ease the work of students during the requirement analysis. At the same time, in many cases, these projects require students to elicit requirements and evaluate their system with real users, something that they have limited experience in doing. This topic requires further research to understand student preferences and experiences in capstone projects. Figure 1 depicts that 55% of the projects are volunteer systems and the rest 45% are compulsory systems. Figure 3 shows that 73% of the projects are product driven and the rest 27% are objective oriented or Goal oriented projects. This result is in line with the results in [17], where the study showed that most of the projects were expected to deliver a product at the end of the course.

License is an explicit field in the project proposal template as shown in Table 1. In the course under study, customers are free to decide under which license they want to release the software that will be produced by the students´ team. By default, we assume the software will be released under an open-source license. From the analysis, we see that in a few cases where the software code or data need confidentiality, the customers specified a more restrictive license. In the year 2021 and 2022 we can see some projects do not specify the license. It is concerning that many customers just accept the default or do not specify the license in their proposals. This might create conflicts in subsequent phases. In general, it is important to increase awareness of customers and students about issues connected to intellectual property rights and different license models.

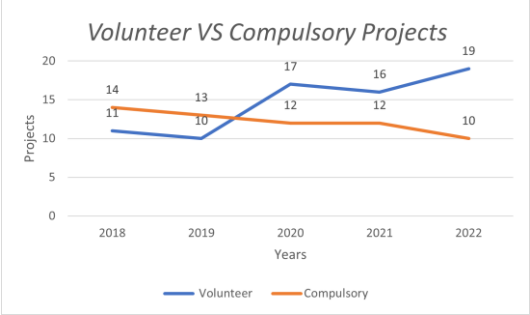

Figure 1 Year wise Projects Volunteer Vs Compulsory

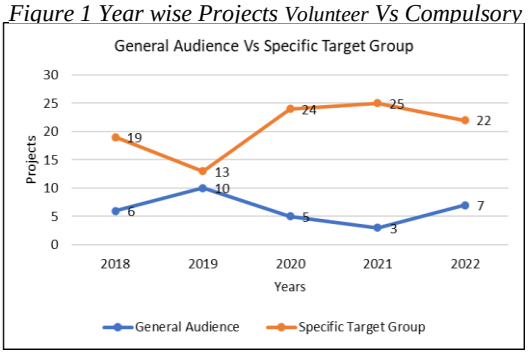

Figure 2 Year wise Projects Targeted for a specific Audience Group Vs General audience.

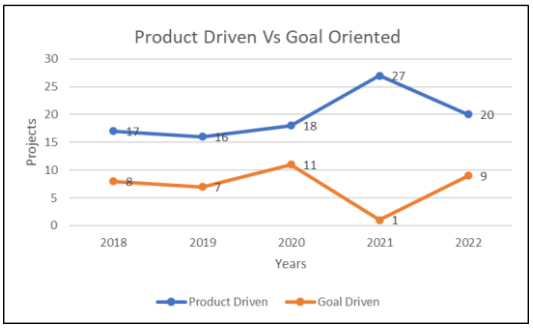

Figure 3 Product oriented Vs Goal Oriented Projects

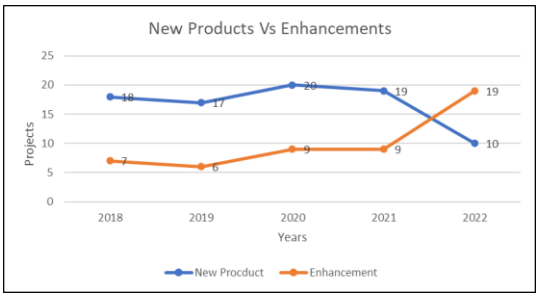

Figure 4 New Product oriented Vs Enhancements Projects

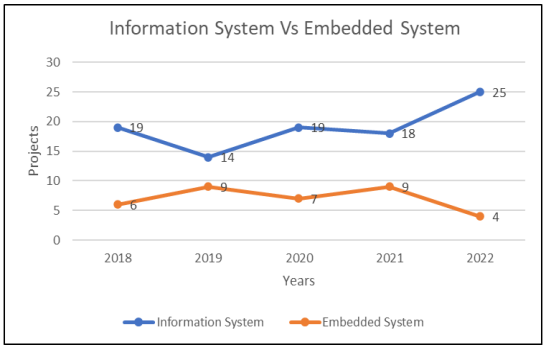

Figure 5 Information System Vs Embedded Systems

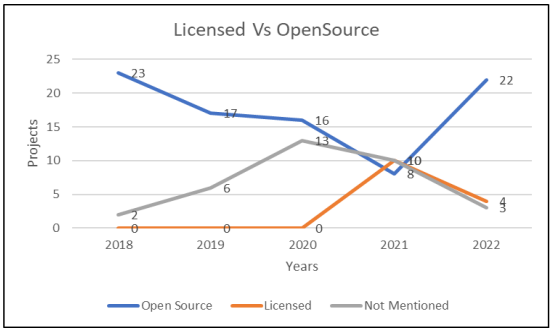

Figure 6 Licensed Vs Opensource Projects

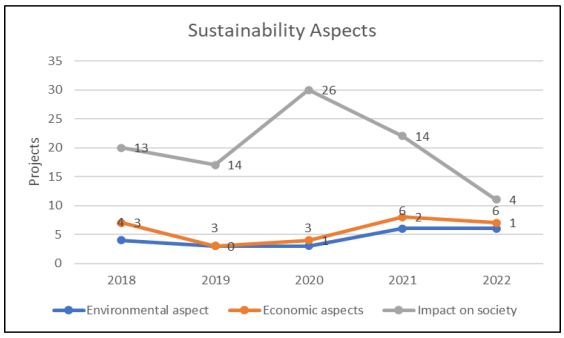

Figure 7 Sustainability Aspects over 5 years

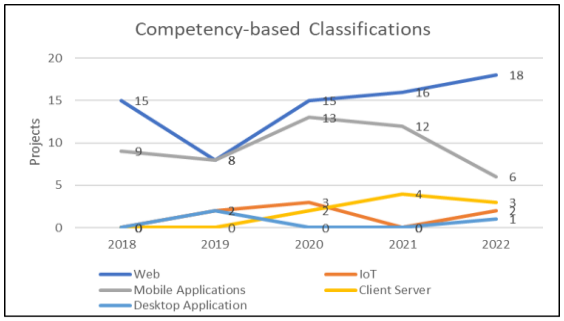

Figure 8 Competency-based Classification over 5 years

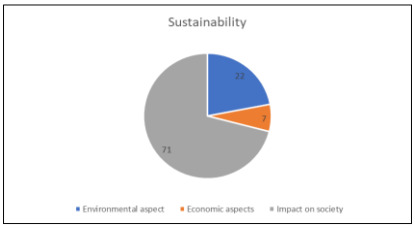

Figure 9 Accumulated results of Sustainability Aspects

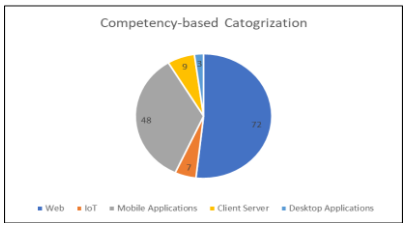

Figure 10 Accumulated result of Competency-based Classification

6 Discussion

6.1 Implications for course organization

The analysis clearly shows the diversity of projects that are of interest to customers. When industry is free to propose projects based on their own interests and needs, the result is a portfolio of projects that use different technologies, covers different aspects of a project life cycle, for different domains, etc. This diversity is important because it provides students with (i) a realistic context for their project and (ii) an increased awareness of the variety of job contexts in which they might be operating. Different projects also give students the possibility to build on their existing skills and/or develop new

ones. However, it is also clear that the heterogeneity of projects brings along several challenges. We do not want here to claim that there is an ideal solution. It is up to the course staff to decide, considering their context, the degree of heterogeneity that is appropriate for their context. The proposed classification of projects might offer a critical instrument to analyze the portfolio and support in decision making. How similar or different are the projects that are part of our portfolio? Do we offer enough variety to cater for the different interests of the local industry Do we have the competencies among the staff to supervise the projects and then evaluate them in a fair way? For example, the heterogeneity of projects must be considered in setting the evaluation criteria. It is not fair to evaluate the outcome of a project that requires acquiring new technical competences using the same criteria than a project that is using standard technology that students have already learned in their study. Formative and summative assessment of projects is indeed one of the main challenges faced when the degree of heterogeneity increases. Heterogeneity of projects makes it difficult to define clear-cut criteria, making the context blurred for both students and evaluators. For example, it is difficult to define a precise number of points (or percentage of the grade) for requirement elicitation when some projects get most of their requirements pre-defined by the customers, while for others that is the main challenge of the project. The same holds also for other aspects of the project. How much should one weight customer interaction management when some of the groups are dealing with high experienced and unproblematic customers, while others are working in more complex contexts? Addressing these challenges in a successful way requires a high degree of openness and focused effort, towards students as well as evaluators. On one side, it is critical to make sure that students understand the complexity of the evaluation criteria and have a reflective approach to the specific challenges of their projects. In our course, we experienced with both peer- and self-assessment by students at mid-term, when all groups must deliver a draft of their report. Self-assessment seems to be an effective way to help students understand how their work is evaluated. On the other side, it is critical to get a team of committed and highly competent evaluators able to appreciate the challenges that each group has faced. In our experience, this requires long conversations to make sure that each group gets a fair evaluation in the general context of the course. Combining evaluators from academia and industry helps to bring forward different issues. This is on the overall a resource demanding process that requires an explicit commitment from the teaching staff as well as the study program to provide the necessary resources. Heterogeneity of projects might be important to provide students with meaningful and realistic experiences, but it does not come for free.

An issue that also needs attention is connected to licensing of the software produced by students, as discussed in the previous section. In general, we experience that this is a topic about which many students have limited knowledge. This is an important issue for which we need to create an increased awareness because the choice has critical ethical, legal, and technical implications. Whether this is an issue to be addressed at the level of the capstone project course or should be introduced earlier is open to debate.

6.2 Implications for program organization

Capstone projects are an important tool for program organizers to assess if the competencies that are developed in previous years are adequate. Do the courses that are offered in the program prepare students to successfully complete their capstone projects? More importantly, if we look at the capstone project as a proxy of projects that students will face at work, does the study prepare them to different work life experiences? For example, when classifying the projects, it became evident for us a gap in the preparation of the students. As discussed in Section 5, a large number of projects are enhancements on existing projects. This is something that we often experience students struggling with. When they reach the final capstone project, students have worked with different types of projects. However, mainly due to practical reasons, these projects are starting from scratch. Students have therefore no experience with managing and working with others code. So, this study shows that working on existing projects to enhance them is at least equally important as working on projects from scratch, in line with the results in [20]. Therefore, we recommend that, where possible, course learning outcomes (CLO) and/or program learning outcomes (PLO) should explicitly address the competencies needed to ensure the reusability and reuse of software source code. This is also in line with the recommendations for Sustainable Computer Science, highlighting the need to teach students not to waste resources in rewriting existing codes and to promote an open source culture [25].

7 Conclusion and Future Work

This paper studies a course in which students work in teams to develop a product for an external customer. The focus is on the type of projects that are proposed by customers at the beginning of the course. For this purpose, we analyzed the projects` proposals from the past five years. By classifying the proposals, we identify trends that might be used as a starting point to reflect on course and program organization.

From our experience, classifying projects from external customers emerges as an important strategic tool for helping teachers to reflect on their work. The portfolio of projects that any course might propose is largely contextual, depending for example on the network of the teaching staff outside academia and the local actors. It is therefore important to work systematically on the portfolio to understand its strengths and weaknesses. A categorization of projects as we suggest might help to identify not only local trends, but also e.g., the lack of projects of a certain type and start a targeted search for additional projects. In addition, the analysis also helps to reflect on course and program issues. For example, the analysis might identify a level of heterogeneity that is not manageable by the teaching staff. The study can also be used as an input for external customer of capstone project to add the missing but required features like sustainability.

The study is based on a single case. This is a limitation of the research. It is therefore necessary to conduct further research to finalize the proposed classification by applying it in other capstone projects. We also aim at developing more detailed guidelines for helping teachers to categorize their project portfolio as a strategic tool for course and program planning.

Acknowledgements

The work is partly funded by Excited, the Norwegian Center for Excellent IT Education. https://www.ntnu.edu/excited

References

1. Sindre, G., Giannakos, M., Krogstie, B.R., Munkvold, R., Aalberg, T.: Project-based learning in it education: definitions and qualities. Uniped. 41, 147–163 (2018).
2. Groeneveld, W., Vennekens, J., Aerts, K.: Software Engineering Education Beyond the Technical: A Systematic Literature Review. In: Proc. of the 47th SEFI Conference (2019). pp. 1607–1622. arXiv (2019). https://doi.org/10.48550/ARXIV.1910.09865.
3. Khakurel, J., Porras, J.: The Effect of Real-World Capstone Project in an Acquisition of Soft Skills among Software Engineering Students. In: 2020 IEEE 32nd Conference on Software Engineering Education and Training (CSEE&T). pp. 1–9. IEEE, Germany (2020). https://doi.org/10.1109/CSEET49119.2020.9206201.
4. The Joint Task Force on Computing Curricula: Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering. Association for Computing Machinery,New York, NY, United States (2015).
5. Sahami, M., Roach, S., Cuadros-Vargas, E., Hawthorne, E.K., Kumar, A., LeBlanc, R., Reed, D., Seker, R.: ACM/IEEE-CS computer science curricula 2013: implementing the final report. In: Proceedings of the 45th ACM technical symposium on Computer science education. pp. 175–176. ACM, Atlanta Georgia USA (2014). https://doi.org/10.1145/2538862.2538863.
6. Sedelmaier, Y., Landes, D.: SWEBOS – The Software Engineering Body of Skills. Int. J. Eng. Ped. 5, 20 (2015). https://doi.org/10.3991/ijep.v5i1.4047.
7. Groeneveld, W., Becker, B.A., Vennekens, J.: Soft Skills: What do Computing Program Syllabi Reveal About Non-Technical Expectations of Undergraduate Students? Presented at the ITiCSE '20: Innovation and Technology in Computer Science Education , Trondheim Norway June 15 (2020). https://doi.org/10.1145/3341525.3387396.
8. Garousi, V., Giray, G., Tuzun, E., Catal, C., Felderer, M.: Closing the Gap Between Software Engineering Education and Industrial Needs. IEEE Softw. 37, 68–77 (2020). https://doi.org/10.1109/MS.2018.2880823.
9. Cico, O.: Incorporating societal topics in software engineering education: A case study of a customer-driven course. Presented at the (2020).
10. Cico, O., Jaccheri, L., Nguyen-Duc, A., Zhang, H.: Exploring the intersection between software industry and Software Engineering education - A systematic mapping of Software Engineering Trends. Journal of Systems and Software. 172, 110736 (2021). https://doi.org/10.1016/j.jss.2020.110736.
11. Tenhunen, S., Männistö, T., Luukkainen, M., Ihantola, P.: A systematic literature review of capstone courses in software engineering. Information and Software Technology. 159,. https://doi.org/10.1016/j.infsof.2023.107191.

12. Einarson, D., Frisk, F., Klonowska, K.: WORK-BASED LEARNING IN COMPUTER SCIENCE EDUCATION – OPPORTUNITIES AND LIMITATIONS. In: 18th International CDIO Conference. pp. 302–312 (2022).
13. A. Martins, A. Bragança, N. Bettencourt, P. Maio: PROJECT-BASED LEARNING APPROACH IN A COLLABORATION BETWEEN ACADEMIA AND INDUSTRY. In: 15th International CDIO Conference, Aarhus University, Denmark (2019).
14. Oates, B.J., Griffiths, M., McLean, R.: Researching Information Systems and Computing. SAGE Publications (2022).
15. Ogenrwot, D., Tabo, G.O., Aber, K., Nakatumba-Nabende, J.: From Undergraduate (Software) Capstone Projects to Start-ups: Challenges and Opportunities in Higher Institutions of Learning. In: Federated Africa and Middle East Conference on Software Engineering. pp. 73–82. ACM, Cairo-Kampala Egypt (2022). https://doi.org/10.1145/3531056.3542775.
16. Burge, J.E., Gannod, G.C.: Dimensions for Categorizing Capstone Projects. In: 2009 22nd Conference on Software Engineering Education and Training. pp. 166–173. IEEE, Hyderabad, India (2009). https://doi.org/10.1109/CSEET.2009.37.
17. N, D., G, S.: Effective approach in making Capstone project a holistic learning experience to students of undergraduate computer science engineering program. J. Technol. Sci. Educ. 8, 420 (2018). https://doi.org/10.3926/jotse.427.
18. Vitner, G., Rozenes, S.: Final-year projects as a major element in the IE curriculum. European Journal of Engineering Education. 34, 587–592 (2009). https://doi.org/10.1080/03043790903202975.
19. Linares-Vásquez, M., McMillan, C., Poshyvanyk, D., Grechanik, M.: On using machine learning to automatically classify software applications into domain categories. Empir Software Eng. 19, 582–618 (2014). https://doi.org/10.1007/s10664-012-9230-z.
20. Connolly, A., Hellerstein, J., Alterman, N., Beck, D., Fatland, R., Lazowska, E., Mandava, V., Stone, S.: Software Engineering Practices in Academia: Promoting the 3Rs—Readability, Resilience, and Reuse. Harvard Data Science Review. 5, (2023). https://doi.org/10.1162/99608f92.018bf012.
21. Cico, O., Jaccheri, L., Duc, A.N.: Software Sustainability in Customer-Driven Courses. In: 2021 IEEE/ACM International Workshop on Body of Knowledge for Software Sustainability (BoKSS). pp. 15–22. IEEE, Madrid, Spain (2021). https://doi.org/10.1109/BoKSS52540.2021.00015.
22. Purvis, B., Mao, Y., Robinson, D.: Three pillars of sustainability: in search of conceptual origins. Sustain Sci. 14, 681–695 (2019). https://doi.org/10.1007/s11625-018-0627-5.
23. Cico, O.: Incorporating societal topics in software engineering education: A case study of a customer-driven course. Presented at the (2020).
24. Hughes, B.: Software Project Management 5e. McGraw-Hill Education (2009).
25. Fang, Y., Neufeld, D.: Understanding Sustained Participation in Open Source Software Projects. Journal of Management Information Systems. 25, 9–50 (2009). https://doi.org/10.2753/MIS0742-1222250401.