Project Course 2023-2024

Faculty of Science and Engineering
Åbo Akademi University
8.9.2023 – 15.3.2024

Contents

1. Course Overview

The objective of the project course is to **plan, design, implement, and deliver a software product in a team**. The team (I) creates a specification of the product, often with the help of an external customer, (II) creates a project plan (III) designs a solution, (IV) implements the solution, and (V) demonstrates the solution. Also, the business aspects of the product developed are to be considered. All these activities are performed within the time limits of the course (i.e., Study Periods I-III). Students are **evaluated** during the execution of the project **based on the deliverables, presentations, and product demonstrators that they produce.** The final version of the product should be demonstrated to a student project competition, ICT Showroom (https://www.facebook.com/ictshowroom/), towards the end of the course, where the students demonstrate their solutions and compete against other student projects from the three universities in Turku.

In brief, the course goes as follows:

- You form a team of 4 to 6 persons.
- You choose a product to be built (based on your own idea[1] or ideas suggested by external customers)
- You create a project plan describing how to realize the initial product idea into the final product, the number of persons in the team and the length of the course.
- You execute the plan. You meet regularly with your team and work together on the product. You review the plan often and react to the potential difficulties. Finally, you deliver your product by the specified deadline.
- Your present regular status updates on how your work goes, where you reflect about how the actual project execution is in relation to your initial plan.

2. Course Objectives

This course is not (only) about programming. In this course, **you will need and develop many other skills and competences**, including:

- *Interact with a customer* and learn how to communicate requirements and design decisions.
- *Plan and follow up a software development project*. Review the plan during the execution of the project and reflect over your initial expectations and estimations.
- *Work in a team*. Team coordination, meetings, planning and internal communication. Use basic collaboration tools in software development such as an issue tracking system and a software repository and version control system.
- *Carry out a development project from beginning to end*. Create and document a design. Implement the design. Test the implementation.
- *Presentations and documentation*. Present your project, product, plans, and documents to your colleagues, stakeholders, and the general public. Produce documentation that is usable and understandable.
- *Business planning* – reflect on which are the end users of your project, why they would potentially buy your product and how you could potentially commercialize it.

Teams are strongly encouraged and advised to join the events arranged by Boost Turku. Visit their webpage www.boostturku.com and their pages on Facebook www.facebook.com/boostturku.

[1] Project ideas suggested by students have to be approved by lecturers wrt complexity, suitability for the course, etc.

3. Course Requirements

This section describes different process requirements for your project. These are requirements on how you should work and carry out your project. These are the **main requirements** to pass the course:

3.1 Attendance to Lectures and Meetings

Attendance to lectures and meetings is compulsory. We know that things happen in your life; you get ill, you have other courses, you have other things going on. That is why we accept 2 absences for a good reason (being tired in the morning is everything but a good reason…). **Every absence after the first two absences will lower your personal grade by one point.** Attendance is recorded by signing a list that is circulated at the beginning of the meetings.

Also – attending a common meeting means being present from 8.30 to the end. We respect each other's work and efforts by being present!

3.2 Teamwork

You will work **in a team**. It is not possible to pass the course by working alone. The teams are composed of 4-6 persons, and you will form the teams yourselves. However, the lecturers reserve the right to modify the team composition for the benefit of the objectives of the course. If you are not able to find a team, please contact the lecturers in good time.

Teams are expected to distribute evenly their workload between all members. This does not mean that everybody should do everything. Good project planning involves delegating the right tasks to the right person.

Project management requires a fair amount of work and responsibility. Choose the right person as project manager (note: the project manager is not supposed to do everything in the project). See role descriptions in the following sections.

When building your team consider:

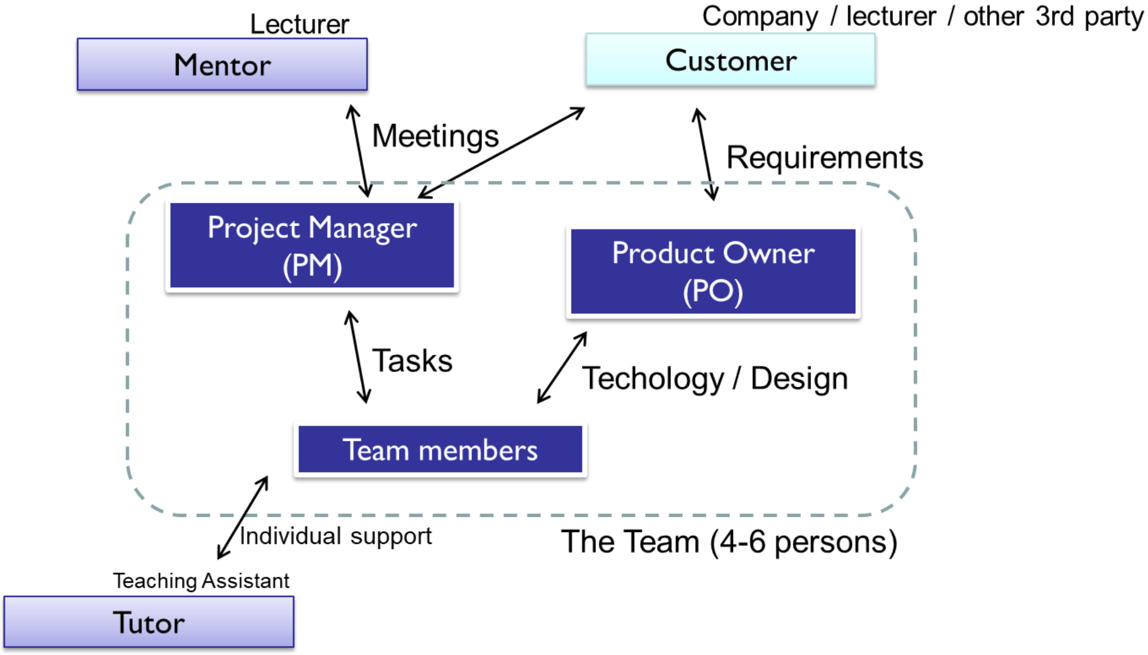
- Choose people with different and complementary skills. Make sure that your team has most of the skills needed to complete the project.
- Discuss when and how you are going to meet and work together. You can meet face to face, online, by email, etc. In any case, you should check that your timetables are compatible.

3.2.1 Team Member Roles

Each project must have an **(External) Customer**. By external, we mean that the customer is not part of the team. In most cases, the customer will be a professional working in the software industry or a university lecturer. In some cases, it is also possible to have a project where the customer is one of the team members. In this case, the team itself will define the product completely, but approval from the lecturers is needed for the resulting specification. This is a suitable choice if you already have a really good idea that you want to transform into a product. Otherwise, working with an external customer is the real experience: you must learn to communicate with another person and to understand her or his vision of the product.

The Team Members can have different roles (responsibilities) inside the team, e.g., requirements analysis, designing, developing, testing, business planning, documentation, which are decided based on the needs of the team. Each team must have two mandatory roles: **Project Manager** and **Product Owner**. Together they represent the interface of the team to the outside world. The **Product Owner** oversees elicitation (negotiation) of the product requirements with the external customer, creating a product vision and validating these requirements. S/he is also supposed to manage the requirements throughout the project, by keeping track of new requirements, implemented requirements, validated requirements, etc. The **Project Manager** oversees planning, coordinating, executing, and reporting the activities of the project. All the other members can have one or several roles and should assist the product manager and product owner with their tasks.

Each team is assigned a **Mentor**, which is one of the lecturers in the course. The mentor is a support person for the team, related to managerial issues. This means e.g., helping to set up the organization of the project, helping with prioritizing, advising on discussing with customers and finding the right activities needed for going forward. The Project Manager should contact the Mentor if s/he feels any need for support in organizing the work of the team (the Mentor is the direct support person of the Project Manager). The team should have **at least 2 meetings with the Mentor**, but can of course have more, if needed. The Mentor is the chairperson of these meetings.



3.3 Executable Demonstrator

There should be an *executable demonstrator* at the end of the project. That is, you need to deliver a running working system, not a plan nor a vision of a system. The demonstrator should be delivered with documentation and access to the source code. The demonstrator should show that the product requirements are fulfilled.

3.4 Use a Backlog Management System

You should have an active backlog all the time. The backlog is the list of requirements/actions/tasks that the team keeps updating during the project, that are at that point in time relevant for achieving the product increments. You can use tools like google spreadsheet, trello.com, aha.io, asana.com, etc.

3.5 Time Tracking

Each team member should report the number of hours used so far in the project, detailed in the following categories: project management, lectures (on Fridays), meetings (with your group), learning and experimenting (new programming languages, tools), analysis, designing, coding, testing, and documentation. You can also use charts to visualize the data easier.

Example:

	Pekka	Anna	Marcus	Total
Management	10	0	0	**10**
Lectures	2	5	6	**13**
Meetings	3	3	2	**8**
Learning	0	0	5	**5**
Analysis	3	0	0	**3**
Design	5	0	0	**5**
Coding	0	15	10	**25**
Testing	0	10	5	**15**
Documentation	1	0	0	**1**
Total	24	33	28	**85**

Add new categories if needed (graphics and music in a game, page design in a web application, etc.). Compare the amount of effort you have put into the course with your own personal expectations. Also, try to balance the activity of the team so that everyone puts in a similar effort overall.

Time-tracking is easy if you use the right tools and do it constantly. MS Excel and Google Sheets are good examples of useful tools.

4. Course Practicalities

4.1 Lecturers

Jerker Björkqvist	jerker.bjorkqvist@abo.fi
Adnan Ashraf	adnan.ashraf@abo.fi
Anna Sell	anna.sell@abo.fi
Ville Harkke	ville.harkke@abo.fi

4.2 Teaching Assistants

Andrei-Raoul Morariu andrei-raoul.morariu@abo.fi will be the Teaching Assistant (TA). His role is

- To assist students on individual level to form the teams
- To assist the teams during the project with practical matters
- To assist with the evaluation of the course

4.3 Course Schedule

We will meet in room **110A&B** in the **Agora** house, on Fridays. The meeting times are available at: https://abacus.abo.fi/proj.nsf

The time reserved for meetings is **from 8.30 to 12.00.** We may finish earlier than 12 o'clock, but you should reserve 3.5 hours for the Project Course meetings. **Attendance to these meetings is compulsory**: all the course participants should attend these meetings.

4.4 Registration and Course Web Page

Please register for the course in https://student.abo.fi portal.

The course website is at https://abacus.abo.fi/proj.nsf. You should use this site to upload your course deliverables.

4.5 Preliminary Knowledge

This course is mainly targeted at Master level students in Computer Engineering, Computer Science, and Information Systems. A Bachelor level degree or equivalent is a requirement for participating. We encourage cross-domain projects, so students from other areas are also welcome. We also encourage that the teams consist of students with different and complementary backgrounds (e.g., graphic design, project management, development, testing, business, marketing, etc.).

4.6 Literature

We do not follow any specific course book. However, there are many books on software engineering and software project management that can help you in this course. We recommend checking the following books:

- Roger S. Pressman, Bruce R. Maxim, Software Engineering: A Practitioner's Approach, 9th Edition, 2019.
- Hans Van Vliet, Software Engineering: Principles and Practice, 2008.
- Frederick P. Brooks, The Mythical Man-Month: Essays on Software Engineering, 1995.

Material for the exploitation/business plan development (see Section 5.6) can be downloaded from the following page: http://www.businessmodelgeneration.com. Material regarding entrepreneurship and startups is also here http://steveblank.com/.

4.7 Evaluation

This course corresponds to **15 ECTS** points. Business students can opt to do a **5 ECTS** version of the course. Please note that 1 ECTS point corresponds to roughly 27 study hours. That means you should expect to allocate to this course around 405 hours of work.

There is no examination in this course. Instead, you (and your team) should create and deliver ***several deliverables on time***, as described below.

Your work will be graded from 0 to 5 (0 –fail, 1-5 –pass). Your course grade will be based on your active participation in lectures, your presentations, project implementation, and project deliverables. Course grades are individual. However, the baseline for the evaluation is the common evaluation of the team, where individual contributions are considered. The evaluations are based on the opinions of the course lecturers and external experts.

The final grade is based on three evaluations:

- 1st evaluation – Phase 1 - 10%
- 2nd evaluation – Phase 2 - 30%
- 3rd evaluation at the end of the course - 60%

You need to pass all three evaluations successfully in order to pass the course. The final course grade will be the weighted average of the 3 evaluations. In the evaluations, the following will be regarded: project idea, team setup, actual system, project plan (project description, schedule, risk analysis, quality), user guide (clarity, usefulness), technical documentation (design structure, clarity), business/exploitation plan, and the quality of the presentations made. **Creativity, innovative solutions, and problem solving** are especially appreciated.

5. Course Deliverables

All deliverables should be uploaded onto the Project Course webpage. Templates with additional information about each deliverable (except business plan) can be found under the *templates* folder at https://abacus.abo.fi/proj.nsf. All deliverables should be uploaded on the course web page according to the schedule specified in Section 4.3 as documents that can be opened in Microsoft Word or as PDF.

5.1 Project Status Report Presentations

Teams should regularly report the status of their work via status report presentations as indicated in the schedule of the course.

The presentation slides are expected to be uploaded on the project server https://abacus.abo.fi/proj.nsf by each team **before** any scheduled presentation listed above. Although the slides are not regarded as deliverables in the context of the project, they are mandatory for passing the course. For material on what the slides should contain for the various status reports, please read the next section.

A status report presents a clear and sincere view of how the project is going. Create these status reports for your own sake: The objective is that **you** measure the status of your project and determine the next steps accordingly. If for any reason your project has not advanced as planned, you should take any necessary action to put the project back on track.

The project status report should include the following: **planned work**, **achieved work, next steps, challenges, and deviations from plan**. You should first gather information in the categories listed below (Sections 5.1.1 – 5.1.3) and then explain the achieved work via time tracking and completed work; the next steps via the planned work, the challenges via risk monitoring, and the deviations from plan via project summary.

The status report should be a snapshot of the current state of the project. But you should continuously keep track of requirements, planned work, task allocation, time allocation, etc. Do not do it just to keep the lecturers happy but for coordinating the project and knowing where you are at any given moment.

5.1.1 Completed Work

Review the tasks in your original project plan. For each task, determine if the task is ready, pending but still on schedule or delayed. Indicate the progress of each task (percentage). If delayed, indicate when it was due.

Example:

Task	**Status**
T3 Design document	ready
T1 Web interface	on schedule (10% ready)
T2.1 Database Schema	delayed (90% ready, due 1 week ago)
T2.2 Database Implementation	delayed (0% ready due 2 days ago)

5.1.2 Planned Work

For each pending task (on schedule or delayed), try to determine the amount of work (effort) that will require completing the task. Compare it with the original effort estimation of the tasks (according to the original project plan). Finally, decide if you can really complete the task by the final course deadline or if you need to review the task (and the project plan).

Example:

Task	**Status**	**Effort Estimation**	**Original Effort Estimation**	**Needs Review**
T1 Web interface	on schedule	10h	10h	N
T2 Database Schema	delayed	5h	1h	N
T2.2 Database Implementation	delayed	20h	10h	Y
Total		35h	21h	

5.1.3 Risk Monitoring

Review the risks that you identified in the project plan. Plan the necessary actions to avoid these risks becoming a reality or, if that is the case, review your project plan.

5.2 Deliverable: Project Plan

The project plan document describes how you are going to execute your project producing your product.

You will need to include:

- A general description of the planned product (**product vision statement**)
- Team composition and roles. Who is responsible for doing what?
- During the project, you will find yourselves performing various activities, such as planning, analyzing requirements, designing your product, implementing, assessing the quality of your product, etc. These activities most often overlap with each other. In order to be in control of what needs to be done and when, we require that you set up **milestones (dates)** and **tasks** specific to your project, and **schedule** them throughout the duration of the course.
- Quality assurance plan. How you plan to ensure that the product meets the requirements.
- Project risks. Describe the main project risks, their avoidance or mitigation strategies.

Advice

- While planning, you should take into account other courses and vacations.
- Give yourselves some slack time.

5.3 Deliverable: Technical Documentation

The technical documentation describes how you are technically going to achieve your product. It is composed of three parts: requirements, design, and implementation. There will be several versions of this document to be delivered throughout the course. Each version is focused on a different part of the product.

In the requirements part, you need to detail the functionality of the product, its user interface, as well as the non-functional properties of your product.

In the technical design part, you need to describe the architecture of the system, the static as well as dynamic aspects in your product, the design of the user interface, and the potential databases.

We recommend you use Service Blueprints as the way to describe the system that you are building. You will get information on how to use Service Blueprint during the lectures. Some examples can be found on the following webpage.
https://miro.com/guides/service-blueprints/

In the implementation part, you need to describe the technical realization of your project. Implementation languages, frameworks, libraries, hardware components, etc.

Advice

- Document the most important aspects of your design. Ask yourself what you would like to read about your design if you would need to modify your software two years after it has been delivered.

5.4 Deliverable: Prototype

This should demonstrate the progress you have made to your product by the end of Phase Two. It could be a mockup of your future product; it could be a set of slides: your choice. The Business Pitch Panel and the lecturers will need to understand what your product is about from the demo of your prototype. Think of it as the rehearsal before the ICT Showroom.

5.5 Deliverable: User Guide

This document describes how to install and use your product. Focus on the most important information that your **end user needs to know for using the product.**

Sections

- Product description
- Technical features
- Installation instructions
- Basic usage instructions
- Troubleshooting

Advice

- Do not document what is obvious ("Use the open command in the file menu to open a file").
- Ask yourself as a developer, what do you need to necessarily explain to or warn about your future customers.
- Ask yourself what you would need to read, if you are a user who just bought your product.

5.6 Deliverable: Exploitation/Business Plan

Create an exploitation/ business plan describing how the product can be used as the basis for a profitable business.

Use the Business Model Canvas at https://www.strategyzer.com/library/the-business-model-canvas as a basis for your Business Plan. On the books page https://www.strategyzer.com/library?type=Books you can find several books explaining how to use the model. The Business Model Canvas helps you develop your arguments for the business pitch deliverable in December. For the final business plan deliverable (March), develop your business canvas into a full Business Plan document. You can download a business plan template and see examples of business plans on e.g. https://www.princes-trust.org.uk/how-we-can-help/tools-resources/business-tools/business-plans

The purpose of the exploitation plan is to make you, as a team, think about how you would launch your product on the market. It can be in the form of a business plan, marketing plan or development plan. Whatever future plans you have for your project, the exercise here is to sell, present, and convince an audience to believe in your idea and start using it. You should think: “How will I gain from my product?”. The audience can be the panel on the Business idea pitch investors, customers, and partners, the visitors at the ICT Showroom, or family and friends.

5.7 Deliverable: Poster

The poster will be used at the ICT Showroom in March. Create a poster describing your project. The poster should have three kinds of information: your names, a short description of the goal of the project, and the system and a technical description of your solution. The poster should focus on the most important features and ideas. You should not have too much text.

Prepare the poster using a drawing program and create a PDF-file. Then upload it to the course web site. We will print the posters on A0 sheets. You will get the poster for own use after the event.

Upload this document as a PDF document with the name projectname-poster.pdf onto the course web page (https://abacus.abo.fi/proj.nsf).

5.8 Deliverable: Retrospective Analysis

The last document that you should create is a 2-page document where the whole team compares the project plan with the actual project execution and summarizes the execution of the project.

Sections:

- What went right? Things that went as planned, worked better than expected or you are especially proud of.
- What went wrong? Problems that you needed to solve, or you could not solve. Tasks that took more work than expected or that yielded worse results than expected.
- Lessons learned. What would you do differently if you took the course again?

You should write this document when all the other deliverables are ready. The whole team should meet and participate in the creation of this document. Write the document for your own sake: Look back in time and reflect on what you have learned during this course, shortly but sincerely.

5.9 Deliverable: Source Code

All student projects must be versioned in Git Hub Classroom. Once you create a team, please visit https://classroom.github.com/g/8QL8eNwH and create a team repository. The repository is private, only the team members and the lecturers have access to the files. A GitHub tutorial for beginners can be found here. https://guides.github.com/activities/hello-world/ (Start from Step 2).

5.10 Software License

You own the work you perform on this course. However, the lecturers and other course participants must be able to inspect the source code and execute the program for evaluation purposes. Ideally, we would like to keep your code for future editions of the course. If you want to explore different licensing options, you can follow this link http://creativecommons.org/choose/.

6. Resources During the Course

6.1 Equipment, Development Tools, and Use of Laboratories

Working facilities:

- The meeting room (110A&B) is reserved for the project course all Fridays 8-12. Therefore, when we do not have common meetings, your team can use this room for meetings etc.
- You can also use the meeting rooms and common spaces on the 3rd floor for meetings and working on your project, when they are not used for scheduled activities by the staff
- If you need special hardware (boards) or lab equipment (e.g., for soldering), contact the lecturers

6.2 Third-Party Software

You can use third party software in your project. However, you must:

- Clearly identify the source of the third-party code, library etc.
- Make sure that you, the lecturers, and your colleagues have the right to examine and use that software.

6.3 Unresolved Disputes and Force Majeure

Teams are expected to resolve their own disputes internally. If you cannot reach an agreement, you can bring an unresolved dispute to your Mentor.

This course requires a large amount of work during the whole academic year. It is possible that a team member becomes ill for an extended period of time, quits, etc. In case of force majeure, the mentor/course lecturers should be notified as soon as possible of the issue. However, the remaining team members should not give up. You should review your project plan and try to deliver the best possible results according to the new circumstances. The lecturers will consider this when evaluating your work.

7. Final Thoughts

This course IS:

- An opportunity to work in a team and to have a lot of fun while working hard.
- An opportunity to show your creativity, to work on the project that you really want.

This course is NOT:

- A programming course. You should already know how to program (unless you are a business student!). Also, you will do many other things besides programming.
- A course for people who only want to work alone

How to enjoy this course:

- Choose a good project topic that you consider interesting. Do not be overambitious.
- Work in something small that you can complete and see it finished.

- Find the right team. Ideally, you feel comfortable working together, have compatible timetables, have the same aspirations for the course and have complementary skills.
- Work regularly, ideally some hours each week. Think of this course as a marathon, not as a sprint.

Some final advice:

- Do not be overambitious. It is better to do a small product that does something concrete and well defined than to plan for a larger product that will never be finished on time.
- Make sure that your team has the necessary skills to implement the product.
- Make sure that you will have access to all the necessary tools and equipment required for the project.
- Be innovative. Combine your ideas, your skills, and your ambitions in a new way.