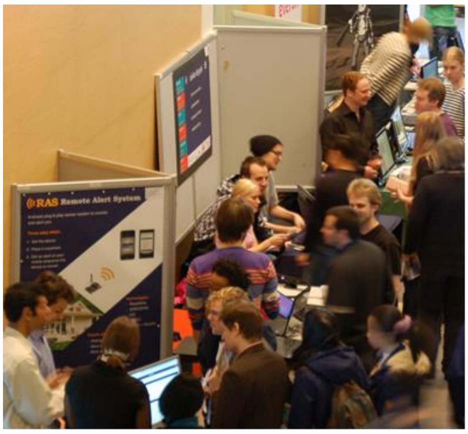
Project Course 2016-2017

Faculty of Science and Engineering
Åbo Akademi University
9.9.2016 – 17.3.2017



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, and (iv) implements the solution. The project should be carried out within a strict time limit. Students are evaluated during the execution of the project based on the deliverables and presentations they produce. There is a project competition, ICT Showroom, towards the end of the course, where the students demonstrate their running solution.

The overview of the course is as follows:

- You form a team of 4 to 6 persons.
- You choose a product to be built.
- You create a project plan describing how to realize the initial product idea into software by $n \in \{4,5,6\}$ persons during six calendar months and a "budget" of n*10 credit units.
- 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.
- You reflect about how the actual project execution went in relation with your initial plan.

There are **two main requirements** to pass the course:

Requirement 1:
There should be an *executable demonstrator* at the end of the project. That is, you need to deliver a running system, not a plan nor a vision of a system:

- It should be delivered with documentation and access to the source code.
- The demonstrator should show that the product requirements are fulfilled.

Requirement 2:
There should be an *exploitation plan* detailing why and how the solution is useful and/or can be turned into a viable business.

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.

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.

1.1 Registration and Web Page

The course web site is at https://abacus.abo.fi/proj.nsf. You should use this site to upload your course deliverables. A participation list is circulated during all meetings. Be sure to **sign the list on each meeting**.

1.2 Lecturers

- Jerker Björkqvist jerker.bjorkqvist@abo.fi
- Mats Neovius mats.neovius@abo.fi
- Karl Rönnholm karl.ronnholm@abo.fi
- Annamari Soini soini@abo.fi
- Dragos Truscan dragos.truscan@abo.fi

1.3 Preliminary knowledge

This course is mainly targeted to master level students in Computer Engineering, Computer Science and Information Systems. Bachelor level degree or equivalent is a requirement for participating. We encourage cross-domain project, so students from other areas are also welcome.

Computer Science, Computer Engineering students: In order to participate in this course you should be familiar with the main concepts in software engineering and have good programming and software design skills. At Åbo Akademi, you can learn this in the following courses: *programmering (programming)*, *programvaruproduktion (software engineering)* and *praktikum i programvaruproduktion (software engineering in practice)*.

Information Systems students: To participate in this course, you should be familiar with the main concepts in information systems theory, investment and financial planning instruments, business models, customer value, project management skills. At Åbo Akademi, you can learn this in the following courses: *analys och design av informationssystem (analysis and design of information systems)*, *investeringsplanering (investment planning)*, *finansiell planering (financial planning)*, *managing the digital enterprise*, *programvaruutveckling och –projekt (information system and software development projects)*, and *user-centered design of information systems*.

1.4 Literature

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

- Roger S Pressman, Software Engineering: A Practitioner's Approach, 2004.
- Hans Van Vliet, Software Engineering: Principles and Practice, 2008.
- Frederick P. Brooks, The Mythical Man-Month: Essays on Software Engineering, 1995.

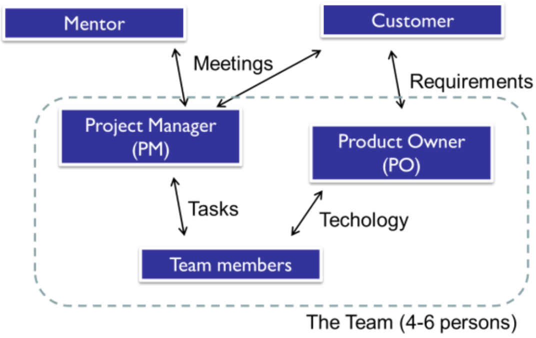
Material for the exploitation/business plan development (see Section 3.5) can be downloaded from the following page: http://www.businessmodelgeneration.com. Material regarding entrepreneurship and startups also here http://steveblank.com/.

1.5 Team Member Roles

Each project must have an **(External) Customer**. By external, we mean that the customer is not part of the development 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 the 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 realize. Otherwise, working with an external customer is the real experience: you have to 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., 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 **Project Owner** is in charge of elicitation (negotiation) of the product requirements with the external customer, creating a product vision and validating these requirements. The **Project Manager** is in charge of planning, coordinating, executing and reporting the activities of the project.

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 setting up the organization of the project, help on prioritizing, and finding the right activities needed for going forward. The project manager should use 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.



1.6 Evaluation

This course corresponds to 10 ECTS points. There is no examination in this course. Instead, you (and your team) should create and deliver on time a number of deliverables described below.

Your work will be graded from 1(0) to 5. 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 after individual contributions are taking into account at the final personal grading. The evaluations are based on the opinions from the course lecturers and external experts.

The course includes 3 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 presentations made. **Creativity, innovative solutions, and problem solving** are especially appreciated.

Software License. You own the work you perform in this course. However, the lecturers and other course participants have to be able to inspect the source code and execute the program for evaluation purposes. Ideally, we will 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/.

1.7 Timetable and Deadlines for Deliverables

NOTE! **Colour code on next page**: green = lecturers lead, students do not need to prepare; red = students present, **students need to prepare beforehand**; blue = ICT Showroom deadline; yellow = holiday

Phase	Week	Date		Agenda	Who talks	Presentation	Deliverables
Phase 1	36	**4.9.2015**	🟩 **L1**	**Introduction**	Lecturers		
	37	**11.9.2015**	🟩 **L2**	**Team building and project ideas**	Lecturers Students		
	38	**18.9.2015**	🟩 **L3**	**Software Process essentials** **Team building and project ideas**	Lecturers Students		
	39	**25.9.2015**	🟩 **L4**	**Lessons Learned** **Team building and project ideas**	Previous Teams Lecturers Students		List of Teams
	40	**2.10.2015**	🟥 **L5**	**Kick-Off** **Mentor/feedback-meetings**	Teams	• Project vision • Product requirements • Project plan (including resources, general schedule) → 10 min/team	• Initial project plan • Tech Doc v. 1 - Product requirements
	41	**9.10.2015**	🟥 **L6**	***Kick-Off+*** (First evaluation)	Teams		• Detailed proj. plan
Phase 2	42						
	43						
	44						
	45	**6.11.2015**	🟥 **L7**	**Project Status Report**	Teams	• Project Status report • Design, Requirements → 10 min/team	• Tech Doc v. 2 - Product requirements - Design
	46						
	47	**20.11.2015**	🟩 **L8**	**Pitching School and Business Aspects**	External speakers Lecturers		
	48						
	49						
	50	**11.12.2015**	🟥 **L9**	**Business Pitch and Demo** **@ Boost Turku**	Teams	• Exploitation/ Business idea → 4 slides, 4 minutes • Demo	• Pitch • Prototype
	51	18.12.2015		*2nd Evaluation Round*	*lecturers only*		
	52			Merry Xmas!			
	1			Happy New Year!			
Phase 3	2	**15.1.2016**	🟩 **L10**	**After Christmas Wake Up**	External speakers		
	3						
	4	31.1.2016	🟦	ICT Showroom registration deadline			
	5	**5.2.2016**	🟥 **L11**	**Project Status report**	Teams	• Project Status report • Product release plan → 10 min/team	• Tech Doc v. 3 - Product requirements - Design - Implementation
	6						
	7						
	8						
	9						
	10	**10.3.2016**	🟥	**ICT Showroom**	Teams		Poster Product Demonstration
	11	**18.3.2016**	🟥 **L12**	**Retro meeting**	Teams	• Final presentation → 10 min/team	• Final product • Final Tech Doc • Business Plan • User guide
	12	21.3.2016		*Final Evaluation Round*	*lecturers only*		

We will meet in room 110A&B in the Agora house. The time reserved for meetings s **from 8.30 to 12.00.** We may finish earlier than 12 o'clock, but you should reserve four hours for the Project Course meetings. **Attendance to these meetings is compulsory**: all the course participants should attend these meetings.

A set of presentation slides are expected to be uploaded on the project server 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 below.

1.8 Project Status Report Presentations

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: **achieved work, next steps, challenges, and deviations from plan**. You should first gather information in the categories listed below (sections 1.9.1 – 1.9.5) 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.

1.8.1 Time tracking

Each project member should report the number of hours employed 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.

Example:

	Pe	Anna	Marcu	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.

1.8.2 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)

1.8.3 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 for 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	

1.8.4 Risk monitoring

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

1.8.5 Project status summary

Resume in 3 bullet points the current status of the project: what is going well and what is not going so well. Add some discussion points or questions if needed.

Advice

- Create the status report to track the progress of your project. Don't do it to keep the lecturers happy.
- Time-tracking is easy if you use the right tools

2 Course requirements

This section describes different process requirements for your project. These are requirements on how you should work and carry out your project.

2.1 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's why we accept 2 absences for a good reason (being tired in the morning is everything but a good reason…). **Every following 2 absences will lower your personal grade by one.** Attendance is recorded by signing a list that is circulated in 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!

2.2 Team work

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 in 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).

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 Course Deliverables

All deliverables should be uploaded in 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 1.7 as documents that can be opened in Micrsoft Word or as PDF.

3.1 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 the responsibility 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 for your project, and **schedule** them throughout the duration of the course.
- Testing plan. How you plan to ensure that the software 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.

3.2 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.

In the implementation part, you need to describe the technical realization of your project.

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.
- Use UML or other graphical language when needed, but also write some text explaining the diagrams.

3.3 Deliverable: Prototype

This should demonstrate the progress you've 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 Demo Day Jury jury 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.

3.4 Deliverable: User guide

This document describes how to install and use your product. Focus on the most important information that your user needs to known.

Sections

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

Advice

- Don't 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'd need to read, if you are a user who just bought your product.

.

3.5 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 http://www.businessmodelgeneration.com as a basis for your Business Plan. Watch the video under *Canvas* on the same page. Under *Book* you can find a pdf file explaining how to use the model.

The purpose of the exploitation plan is to make you, as a team, think about how you would launch you 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, partners, the visitors at the ICT Showroom or family and friends.

4 Other Deliverables

4.1 Deliverable: Poster

The poster will be used at 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 in the course web site. We will print the posters on A1 sheets. We will keep the poster and use it in future editions of the course.

Upload this document as a PDF document with the name projectname-poster.pdf to two websites:

- to the normal course web page (https://abacus.abo.fi/proj.nsf)
- to the ICT Showroom webpage (www.it.abo.fi/ictshowroom)

4.2 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 together and participate in the creation of this document. Write the document for your own sake: Look back in time and reflect what you have learned during this course, shortly but sincerely.

5 Resources during the course

5.1 Equipment, Development Tools and Use of Laboratories

Working facilities: Agora house's (and later, Agora's) premises are available; talk to your team mentor
IT-resources: we provide GIT, Trac.

- For requests please email Karl Rönnholm (karl.ronnholm@abo.fi)
- The meeting room (110A&B) is reserved for the project course all Fridays. So, when we do not have common meetings, your team can use this room for meetings etc.

5.2 Use a Backlog Management System and Source Code Repository

You should have an active **backlog** all the time. The backlog is the list of requirements / actions 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.

Also, you should use a source code repository to keep and share all the assets in your project. The repository is a centralized storage of developed code, but you can also choose to put other documents there. It keeps track of different versions of the files, who wrote what and when. Use the repository as your day-to-day tool.
The most common used tool for code repository management used today is GIT. GIT is free and open source, there are several CLI and GUI clients available. For remote repositories, gitlab.com och github.com are commonly used.

You are free to use the development environment that is best for your project. Use of specific software development tools, laboratories, etc. may be arranged if needed.

5.3 Third-Party Software

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

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

5.4 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.

6 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 in the project that you really want.

This course is NOT:

- A programming course. You should already know how to program. 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. Don't 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 in the project.
- Be innovative. Combine your ideas, your skills and your ambitions in a new way.