# IT2901
# Project report guidelines

This document is made available on the itslearning it2901 course page. All the templates referred to in this document are also available on the course page, together with an overview of relevant deadlines.

Reports have to be delivered via its learning (see Assignments folder). The final version of the report must also be delivered in printed form, with code on USB key/CD.

## General information

You MUST hand in three versions of the project report to your supervisor as specified at course start. (In addition, you might also deliver preliminary versions for feedback at any time upon agreement with supervisor.) Each version is a refined and elaborated version of the previous one. You will receive thorough feedback on the preliminary and midterm versions, and in further work on the report you are expected to follow the advice you receive.

If you are in doubt about the requirements for each version as specified in this document, ask your supervisor in time!

The report should be written in English, though user interface and user manual might be in Norwegian if so required by the customer.

### *Format and Size of the report*

Each report should be clearly identified and include a first page with: identifier of the group, list of group members, date or version number.
Pages should be numbered. We have no absolute limit to the number of pages in the final version. A 'ballpark figure' is 50-70 pages in the main part of the report. More than 70 pages is a sign that you should probably be more concise and reader-friendly.

### *Appendices*

Appendices should be included (only) to provide useful background information and elaboration on points in the main part of the report. Every appendix should be referred to in the main part of the report.

### *Chapter structure*

You decide on a chapter structure, which should be in place in the preliminary version. Having a look at the guidelines for all versions of the report (in this document) will help you decide on an appropriate structure. Having a look at other project reports might also be useful.

### *Models/diagrams*

Generally, make use of appropriate models/diagrams whenever they serve to clarify your points and in line with the standard usage of analysis and design models in software development. Every diagram should be accompanied by a textual description, and a legend should be included for all but mainstream notation.

In some cases, we explicitly *require* the use of diagrams (specified in the below guidelines).

If you are in doubt about which type of diagram to use, ask your supervisor.

### *References*

The report should include a list of references to literature, websites and other resources on which the contents of the report are based.

### *Deliveries (preliminary, mid-term and final version)*

The report with appendices should be handed in as *one* document. That is, an electronic copy is *one* PDF file. A paper copy is *one* booklet.

# Preliminary version of the report

This version is intended to help you get started with important aspects of the project work. As regards analysis and design, you are not expected to provide elaborate accounts in the report; some initial thoughts and sketches will do. However, you should demonstrate in your report that you have come to a mutual understanding with the customer about the project task (including its motivation and scope), and that your team has thoroughly worked on the planning and organization of the project.

We want you to create a chapter structure for the whole report and start filling it with contents. You may modify the structure later.

Contents:

- A description, in your own words, of the problem that you are going to solve for the customer. It might be useful to provide a high-level description of the current situation (AS-IS) and of the target situation (TO-BE).
- A high-level description of the main requirements (functional and non-functional) for the system to be developed. The level of details depends on the type of project and the approach you are choosing. It is however important that you work to reach an agreement with the customer.
- A brief outline of alternative solutions. The relevance of this point depends on your project. In general, you are expected to check if there are already existing solutions available on the market or as open source and make an evaluation of them. When you are evaluating different alternatives, it is important to make clear your evaluation criteria. Evaluation should be discussed with the customer and any choice should be agreed upon with the customer.
- A tentative outline / sketch of the architecture of your solution.
    - We do not require much here, but make a try! ☺
- A description of your team organization
    - Roles and responsibilities
- A justified choice of process model for the development work
    - Will you work iteratively and/or incrementally, will you make a mock-up or prototype; phases and iterations…
- A preliminary overview of the development environment
    - Which tools/infrastructure do you plan to use for development and collaboration, e.g. for programming, versioning, testing, documenting, archiving, communicating within your team, communicating with other stakeholders,..
- A tentative time-plan of the project

## Midterm version of the report

In this version, you should be able to present all of the analysis and most of the design of your system. Some details such as the inner workings of particular modules may be unresolved, but the architecture of your system should be fixed. Having reached an agreement with your customer about all the system requirements, functional and non-functional, is vital at this point. In the mid-term version of the report, you should account for how you will meet the requirements during the remaining course of the project.

Contents:

- The refined / elaborated version of the contents in the preliminary version, with requirements (functional and non-functional) described in detail. (Keep in mind: in the final report, for every requirement, you should be able to say, based on your test results, whether it has been met or not.)
- System architecture - complete and described both graphically and in words. Remember to describe both the logical architecture, e.g. layers and packages, and the physical deployment.
- The design of the components of the system[1], documented by suitable models/diagrams (e.g. data models, class diagrams, sequence diagrams). A very limited amount of source code may be included to illustrate particularly important parts/aspects of the system.
- Test plan, describing the strategy for testing on all levels (unit testing, integration testing and system testing (including acceptance testing))
- Project management documented: possible deviations and how they have been handled, examples of activity plan and status report. Major lessons learned so far.

We expect you to take your supervisor’s feedback on the report seriously. The feedback will help you modify your approach (if necessary) in order to get a good result.

[1] The design of every component might not be complete at this point.

## Final version of the report

In this version, you provide the complete documentation of your project, both the product and the process.

!!!Final reflection notes are to be handed in separately, but are considered part of the final report!!!

<u>Contents:</u>

- Same as the midterm version, with revisions and refinements to make all points complete
- Test results, relating to the test plan
- An evaluation of the quality of the system. Importantly: this should be done with reference to system requirements and test results.
- In appendices:
    - Installation guide, the customer (and the evaluator) should be able to install the software you have developed without any additional help from you
    - User Manual
    - *…Any other relevant documentation*

The final version of the report is handed in together with your system (on CD/DVD or USB key).