# Advice on Preparing Reports in Softwareengineering

Paul Fischer, January 2020

## General

There is not a single, fixed form for reports which document software and the development process. The target group, the customer, is an important factor for how the report looks. However, there is a number of topics which should be addressed in any such report. The following takes up these issues and also considers the fact that it is a report for a course at an educational institution.

A general advice is to avoid to just give a manual for or description of the final product. Instead, also explain why the product looks as it is. Document the development process, decisions made on the way, and how you tackled problems.

The points below do not have to (but might) be the headlines/sections of the report. Below, the word "describe" has to be understood as "describe and justify your choice".

Do not use non-standard terms without explaining them or before they are explained. Do not use "slang" terms in the report, i.e., terms which you developed and used in the group.

In order to be able to give individual grades it is necessary that you document, who is responsible for which part/section/chapter. Please to so in the text whenever the author changes, not only in the beginning (like "Jan did 3 and 4, Anne did 2 and 5"). It is **not** enough to say something like "we all contributed to the same extend", but you have to point out an unbalanced workload in the beginning.

Make sure that your names, study numbers, and group number appears on the cover page.

## Report Structure

### Introduction

- What is it about?
- What is your level of ambition, both with respect to the problem and the grade?

### The Problem and a Coarse Analysis

- Describe the problem and the context/domain in which it appears.
- Describe the scope of the project: Which sub-problems will be considered, which will not be considered (restriction).

### The Goals

- (Functional) What are the main goals of the project, what should be the properties of the final product?

- (Non-functional) Usability, performance, maintainability, extensibility, simplicity, security, ....
- (Non-goals) What will not be considered or tackled.
- (Non-project) Learn new technology, learn project management, ....
- What are the success criteria for the project?

**Analysis of the Problem**

This is independent of the type of project (hard-/software) and the HW-platform or programming language (unless enforced by external factors).

- What are the main challenges for solving the problem.
- What are potential sub-problems.
- Are there know solutions to sub-problems.
- Propose solutions to the problem/sub-problems.
- Select the solutions you want to use and justify you choice.

**Design**

- Describe the general architecture of the solution.
- Identify major system components (components, packages, interfaces, ...)
- Describe the technologies/platforms used.

This serves as a guideline for the implementation. Describe possible alternatives and why you have (not) chosen them.

**Implementation**

- Describe the system components.
- Specific algorithms and data structures.
- If applicable, describe detailed class hierarchies or data base schemes.
- Avoid to describe the whole code in detail (this can be done in an appendix and as comments in the code listing), however point out unconventional coding.

Again, describe possible alternatives and why you have (not) chosen them.

**Note:** Sometimes, "Design" and "Implementation" might have a large overlap and cannot necessarily be separated. These chapters should explain how the system is build and how it works, its components interact.

**Test**

- Which test-types were used and why were they chosen?
- Document the results and potential changes because of errors found.

**Project Management**

- Did you use project management paradigms/tools, if not why?
- How did you assign task to the group members and how was the quality checked?
- How did it go, if there were conflicts, how were they handled?
- How did you manage time?

**Conclusion**

- What was achieved?
- Which goals were not reached and what effect does that have on the project?
- What is good, what is not?
- What did you learn, techniques, project management, socially?
- How could one continue (should one continue)?
- Personal evaluation of the project.

**Note:** "Introduction" and "Conclusion" should be written such that they can be understood without knowing the rest of the report.