

CSCM10 Computer Science Project Research Methods

Writing Your Project Specification

Anton Setzer

Talk based on presentations by Markus Roggenbach on writing
third year and MSc Specification Documents

[http://www.cs.swan.ac.uk/~csetzer/lectures/
computerScienceProjectResearchMethods/current/index.html](http://www.cs.swan.ac.uk/~csetzer/lectures/computerScienceProjectResearchMethods/current/index.html)

April 8, 2019

- ① Introduction
- ② Background Research
- ③ Description of the Project
- ④ Project Plan
- ⑤ Conclusion and Bibliography
- ⑥ Quality of Presentation

Relationship to your Presentation

- Your presentation
 - defines a rough area of work
 - reports on background studies performed
 - gives a first idea about what the project will be about.
- The specification document
 - Clearly identifies what would progress the field.
 - Details the concrete product to be produced.
 - Provides a detailed plan of how to produce the product.

From the Coursework Description

- The specification document contributes 50% of the mark for the module CSCM10.
- Approximately 4000 words,
- due on **10th May 2018**;
- you will need to submit an **electronic copy** via **Blackboard/Turnitin**
- Each report is assessed by your supervisor and a second member of academic staff.

From the Coursework Description

- This report should describe
 - any **background research** that you have carried out and
 - state the **motivation** and **aims of the project**.
- It should include a **complete specification of your project**.
- It should **describe** the **project clearly**
- and the **components of the work which need to be developed**.
- An **outline project plan** for the summer should be included.
- This plan should take into account the **development methodology** being used.
- You should provide a **risk analysis** for the project.
- You should view this document as providing the **plan for the work** you expect to carry out over the **summer**.

From the Coursework Description

- A marking Scheme is available on blackboard.
 - It is highly recommended to look at it carefully.
 - Make sure all items in it are addressed, because you will be evaluated by it.

General Structure (Guide Line)

- Title
- Abstract summarising the project
- Introduction
 - Motivation
 - Aims of the project
- Background research
- Description of the project.
 - Including components of the work.
- Project plan
 - Gantt chart
 - Risk analysis
- Conclusion.
- Bibliography.

① Introduction

② Background Research

③ Description of the Project

④ Project Plan

⑤ Conclusion and Bibliography

⑥ Quality of Presentation

Motivation and Project Context

- Should go into the first paragraphs of the introduction.
- Example opening (D Garner, 2010):
Software written today differs from that written in the past in two main ways: it is more complex, and handles more data than ever before. When Windows 95 was released it contained 11 million lines of code, in contrast six years later Windows XP had 40 million lines [1]. This gives rise to a greater need for systematic testing.
- Such an opening is then followed by 2 - 3 paragraphs giving a narrative on what area your project will address

Question your “Motivation” should answer

- To which **area of CS** does the project belong? Why is this area important?
- What is the **state of art** in this area?
- What is the **gap** that you want to **fill** by undertaking the project?
- Why would this **contribution** be an **important** one?
- What **techniques** do you intend to apply?
What is your special angle?
- How will your envisioned result **make the world a better place?**

- Location: end of introduction
- Example (D Garner, 2010):
More concretely, the aims of the project are as follows:
 - *To develop a method of generating test data with trends using genetic algorithms.*
 - *To demonstrate our approach on a case study of credit card fraud detection.*
 - *To integrate our method in Grid-Tool's DataMaker.*

① Introduction

② Background Research

③ Description of the Project

④ Project Plan

⑤ Conclusion and Bibliography

⑥ Quality of Presentation

What it is about

- Note that this is very individual for each project
- **Typical elements:**
 - A detailed **literature review** for the area of your project.
 - An overview on **existing tools** and an assessment of their suitability for your project.
 - You might include examples for running these tools
 - Give an overview over **related work**.
 - What has been done before?
 - How does this differ from your work.
 - A discussion of **techniques** typically applied in the area (e.g. Software quality assurance by walk through inspection, testing, software metrics)

- Example (part of related work) (D Garner, 2010)
In 1999 Chan et al wrote a paper applying genetic algorithms to fraud detection, which will be the subject of our case study. In this work they stated that their scalable black-box approach for building efficient fraud detectors can significantly reduce loss due to illegitimate behaviour. In many cases, the authors methods outperform a well-known, state of the art commercial fraud-detection system [11].
- This is then explained in more detail.

Make Things Concrete

- Present examples - from the literature, or own ones - that **the reader can follow**.
- A good example is “**short**” and **easy to grasp**.
- The focus needs to be on the **point that you want to illustrate**.
- **Develop a story** – don’t just write “A did a, B did b, C did c”.
- What is your angle to look at things?
- The background research shall underpin your view on the area as written up in the “motivation”.

- 1 Introduction
- 2 Background Research
- 3 Description of the Project**
- 4 Project Plan
- 5 Conclusion and Bibliography
- 6 Quality of Presentation

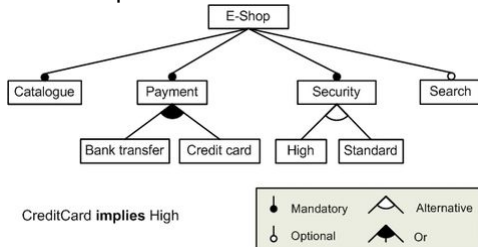
Description of the Project

- Describe in detail what your project is about.
- The reader should get a crystal clear idea about what you are going to create or write about?
 - How does the final product or document look like?

Components of the Work

Feature A distinguishing characteristic of a software item (e.g., performance, portability, or functionality) (IEEE 829).

An example feature model for an E-Shop:



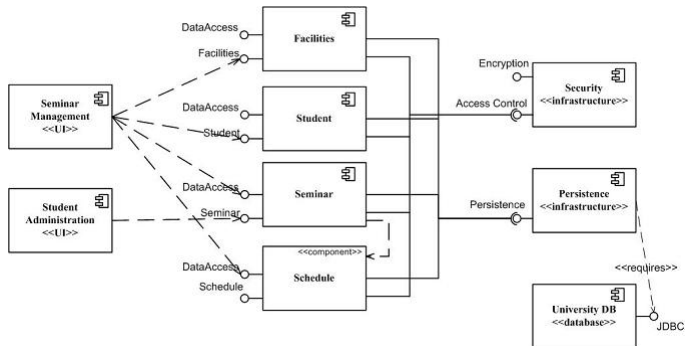
<https://upload.wikimedia.org/wikipedia/commons/thumb/d/d6/E-shopFM.jpg/475px-E-shopFM.jpg>

Components

- **Component** is a unit of composition with contractually specified interfaces and explicit context dependencies only.
- A software component can be deployed independently and is subject to composition by third parties.
- The next page shows an example of a UML component diagram:

*[http://agilemodeling.com/images/models/
componentDiagramUML2.jpg](http://agilemodeling.com/images/models/componentDiagramUML2.jpg)*

Components



- Include
 - Functional requirements
 - Nonfunctional requirements
 - Use case scenarios

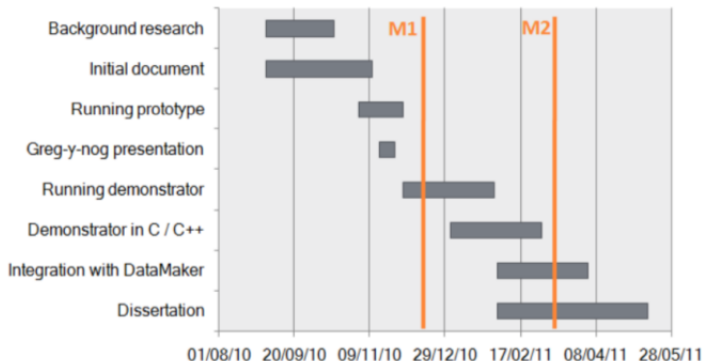
Hard and Software Requirements

- Mostly important for Software and software related projects

- 1 Introduction
- 2 Background Research
- 3 Description of the Project
- 4 Project Plan**
- 5 Conclusion and Bibliography
- 6 Quality of Presentation

Gantt Chart

Example – Gantt chart (D Garner, 2010)



- Each item should be described separately.
- Add project-specific items.

Linking Methodology and Work Plan

- Note SLC prescribe development phases.
- Question: are these phases visible in the project plan?
- At the end of each phase of a SLC, a number of deliverables is due:
programs, specifications, documents.
 - Question: are these reflected in the milestones and deliverables of the project plan?
- Note: D Garner could have done better: milestones M1 and M2 are disconnected from the chosen SLC

- Example - Part of risk analysis (D Garner, 2010)
It may be found that a genetic algorithm cannot solve a more complex data generation problem. We have avoided this by thoroughly researching the algorithms needed and any previous success had by other scientists, but if an insurmountable difficulty does occur we will research deeper into other available methods.
- Elements:
 - What is the bad thing that might happen?
 - How likely is it?
 - How much would it endanger the project?
 - What plan is in place in order to avoid it? (Mitigation).

- You can actually estimate numbers to the probability and the consequences (e.g. on a scale from 1 to 10).
 - The product calculates the risk (product of probability and consequences).
- Make sure that you have not only generic risks (such as loss of data due to hardware failure) but as well **project specific risks**.

Software Life Cycle

- For software and software related projects
- Describe the software life cycle / methodology.
- Say why you have chosen it.
- Example Part of SLC (D Garner, 2010)

We have chosen a Test-Driven development model for this project. This is also known as agile programming and like most models has a sequence of iterations. In contrast to the more usual spiral model which places emphasis on a thorough design being complete before the first implementation, test-driven development has an principle of always having a working, albeit basic version of the software. This is supported by iterations of test-then-code cycles, in which each cycle implements another user feature or piece of functionality [4].

- For non-software centred projects you can investigate research methodologies.
- A small collection of research methodologies is available at <http://www.cs.swan.ac.uk/~csetzer/articles/researchMethodologiesInComputerScience.pdf>

- 1 Introduction
- 2 Background Research
- 3 Description of the Project
- 4 Project Plan
- 5 Conclusion and Bibliography**
- 6 Quality of Presentation

Conclusion

- Summarise what you have found.
- Draw conclusion (a bit of evaluation - more evaluation needs to take place in your dissertation).

- Study **carefully** lecture given in CSCM10 on bibliographies.

- 1 Introduction
- 2 Background Research
- 3 Description of the Project
- 4 Project Plan
- 5 Conclusion and Bibliography
- 6 Quality of Presentation**

Use Spell and Grammar Checkers

- Use a spell checker - we are blind to our own mistakes.
- Use the fact that Word has a grammar checker.
 - Adjust the settings so that it is rather strict on grammar.
 - Often by we click on “don’t show me this error again” which makes Word less strict.

- **Comma, colon, full stop, semicolon**
 - No white space before the punctuation mark,
 - one white space after the punctuation mark.
- **Opening brackets**
 - One white space before an opening bracket,
 - no white space after an opening bracket.
- **Closing brackets**
 - No white space before a closing bracket,
 - one white space after a closing bracket.

Dash

- One white space before a dash,
- one white space after a dash.

More Hints on Writing Documents

- Look at 3rd year project handbook available from Blackboard
 - Student information pages (2017-18 information).

Good Luck with Writing your Specification Document