Guide for the Structure of Your Final Report

Joseph Bloggs [*]
University of Adelaide, South Australia

June 2010

Misquotations are the only quotations that are never misquoted.

Hesketh Pearson

Abstract

Here is where you put a good summary of what your project is. What you have achieved and a hint at how it can be used in future to make life better for everyone.

1 Introduction

A sentence or two describing what your project does is good right at the start.

Here is where you put your project in context. Make this short but expressive. Speak a little about the need or niche that this project fulfills. Use examples, anecdotes, quotes, references to articles to convince the reader (in no more than a few paragraphs) of the need for your application. In this section you can talk briefly about other approaches and why they do not meet all needs - thus your project is needed!

Here is where you can say that your project aims to solve all/part/a-very-small-bit of this problem.

Here is where you say in the remainder of this report is laid out as follows. In the next section I describe, in detail, what my project aimed to achieved. In section 3 I[1] the describe the process and tools I used to meat these goals. In section 4 I discuss what has been achieved and where the goals of this project have been met or exceeded and useful lessons learned from the development process. Finally, in section 5 I conclude with summary of the project outcomes and future work that would enhance its contribution.

[*] Authors' Address: Department of Computer Science, University of Adelaide, Adelaide 5005, Australia. Phone: +61 8 303-8888, E-mail: bloggs@cs.adelaide.edu.au

[1] You can use *I* or *we* to describe the subject considering the concepts in this report, both are OK.

2 Project Aims

Here is where you describe what your project is meant to do. You have already described what the problem is so don't dwell on it here. You just need to tell the reader as clearly as possible what you wanted to build. Don't spend too many paragraphs on this you don't want to rewrite your project plan but you *do* want the reader to understand your vision for what the working project does and what the intended end product is.

3 Approach

This is where you say *how* you built your system. What tools you decided to use and why. You can describe your methodology in terms of your broad milestones (and why they were chosen). You might also describe some of the changes made to your approach but you should emphasise the tools and methods you ended up using so the reader has a clear idea at this point.

4 Results

Here is where you describe what your project achieved in terms of product and in terms of solving problems. If you were able to run trials to test its effectiveness definititely summarise these results here.

You can discuss what you discovered along the way. A lot of this will be about the nature of the problem itself which is interesting to the reader. Some of this might be about little understood or little known features or problems with the software you used. You *should* include things that might be of general interest to a reader that is going to undertake a similar or related project to this one, or might use the same tools. You *should not* include a log book of the troubles you had unless they are of general interest (or just make really good reading).

5 Conclusion

End here with a brief recapitulation of what your project achieved and what needs to be done to verify that it works well and what could be done to extend improve it in future. If there was something amazing discovered about the *process* of completing this project include some words here too (but what you say about the process has to be relevant to other people not just to your experience during the project).

References

[1] B. Bloggs, Being the brother of Joseph is really difficult, *Journal of Blogg Family Relations*, Volume 2, Pages 233-340, 2004, ACM Press, New York, NY, USA.

[2] Venables, A. and Haywood, L., Programming students need instant feedback!, In *Proc. Fifth Australasian Computing Education Conference (ACE2003)*, Adelaide, Australia, Conferences in Research and Practice in Information Technology, Vol 20, pp. 267–272, ACS.

[3] Dalit Levy, Tami Lapidot, and Tamar Paz, It's just like the whole picture, but smaller: Expressions of gradualism, self-similarity, and other pre-conceptions while classifying recursive phenomena, In *13th Annual Workshop, Psychology of Programming Interest Group*, Bournemouth University, UK2001.

[4] Roberts E.S. (1986). Thinking recursively. John Wiley & Sons.

A Some useful stuff that cant fit in the report

put that useful stuff here.