PUSL3190 Computing Individual Project
Final Submission / Defense Guidelines

Final Report - Report structure

The report documents your project's products and the processes that supported their development. It will normally conform to the following structure:

- Title [Front Cover] page (template available on the portal)
- Acknowledgements
- Abstract (see the separate document on writing your abstract)
- Table of Contents
- List of Figures and Tables (if applicable)
- Main body of report
    - Introduction
    - Chapters
    - End-project report (and Recommendations, if applicable)
    - Project post-mortem
    - Conclusions
- Reference List
- Bibliography
- Appendices – in this order please
- User Guide (As a minimum this should indicate how the product can be installed for demonstration, including details of the (minimum) required platform specification. If appropriate, there should also be information on how the system can be used or operated.)
- PID
- Stage plans
- Interim report
- Records of supervisory meetings
- Other material such as designs, preliminary designs, test results, etc.

Main body content

The variety of projects precludes a definitive statement on the content of a project report, but here's one suggestion for the main body:

- Introduction
- Background, objectives & deliverables
- Literature review (if applicable)
- Method of approach
- Requirements
- End-project report
- Project post-mortem
- Conclusions

These are only suggestions, and the contents of your particular report should be discussed with your supervisor.

General advice on content

- Allow yourself plenty of time to write your report and then allow 50% extra on top of that.
- Polish, polish, polish and then polish some more. Don't hand in the first draft (or even the tenth!) Many first/second/third drafts are often dreadful.
- Describe your work to a suitable level. The standard advice is that it should be written at a level that could be read by a (good) final stage student. The best reports display the author's maturity and demand it from their reader.
- Please put some screen shots in your main body … but too many can be tedious.
- You can gain credit for what you did as well as what you produced. The report should therefore describe both your deliverables and the processes by which they were created. Especially consider the activities you carried out that are not obviously visible from the final product. For example, consider requirements elicitation: which methods did you use? Show the reader that you're aware of recognized techniques rather than making it up as you go along: why did you choose them? Describe their enactment and record their findings. What particular problems/challenges did you encounter, and how did you solve them? What parts of your work do you think are particularly worthy of credit: bring these out in your report?
- Do not include significant chunks of code in your report … instead put them on the disk which is attached to the report.

The use of appendices

- The appropriate use of appendices is critical. Firstly, assume that your reader will not read your appendices: in other words, make sure that the main body of the report is a self-contained description of your project (at least at some level of abstraction): If the reader needs to read something in order to understand your project, then put at least a summary of it in the main body.
- Some things will be too voluminous or detailed to put in the main body but may be important to the reader's understanding: put a summary in the main body, and the full-blown version in an appendix. For example, the reader does not want to read pages of test results – put a summary of the testing processes and results in the main body, and the full blown test results in an appendix.
- Material that relates to the wider context (particularly if it is voluminous) is a good candidate for inclusion in an appendix. For example, you might be developing an eCommerce system for a client, and have written a summary describing your client's competitor websites. Generally speaking, the main body needs to focus on your project.

- Another reason for putting material in an appendix is that it is too low level. For example the details of your DB normalisation would normally be in an appendix, with the final schema presented (perhaps as an ERD/LDS) in the main body of the report.
- Another example would be your choice of software development process. You might write a chapter discussing the pros and cons of the different processes available1 , and why you chose the one that you did. But this is not final stage work: put this chapter in an appendix, and put a brief summary in the main body. It is worth noting here that you also need to detail (in the main body) how the process was enacted in your project.
- Ditto for your choice of technologies.
- Another use of appendices can be to include earlier versions of designs: for example initial screen shots – even ones drawn by hand. These can be useful in your description of your process.

Secondly, assume that your reader will read your appendices: The quality of presentation in the appendices should be as good as in the main body. Furthermore, the appendices are not a dumping ground for miscellaneous material, and in particular, all appendices should be referenced/cited in the main body of the report; if they aren't, then you might consider why they are included at all.

End-project report

- An end-project report is produced (say for a Project Board or Client) as part of (and towards the end of) a project!
- Brief summary of the project and its achievements.
- Relist your project's objectives and critically (and ruthlessly) evaluate whether you met the objectives. Projects rarely go perfectly, and an inability to find any real criticism will possibly be met with some suspicion. If your work is for a real client, try to involve them in this evaluation (and include details of their feedback).
- Realisation of business objectives (either to-date or planned).
- Changes made during the project, their reasons and effects.

The post-mortem

A post-mortem is often carried out shortly after a project is over. Looking back (and indeed standing back), you can now evaluate/critically appraise any aspect of the project (although you do not need to repeat any evaluations that were made as part of the project/end-project report). For example:

- Were the project objectives the right ones to adopt?
- Was the product properly specified (in relation to the business objectives)?
- The relationship between the project and the client.

- Did the chosen development process the right one?
- Were the chosen technologies the right ones?
- Your own performance (try to be specific).
- Wider reflections on Client feedback.
- Generally, what lessons are to be learnt for the future.
- • Etc., etc.

Conclusions:

Final, brief, summarising conclusions.

Final Report: Style

- The report should be professionally presented and word processed with (approx.) 1.15 line spacing on one side of good quality A4 paper. Pages should be numbered consecutively except for the title page.
- Sections and sub-sections should employ nested numbering, e.g., Section 1, sub-sections 1.1, 1.2, etc., sub-sub-sections 1.1.1, 1.1.2, etc. This is really important in helping the reader to follow the structure of your report.
- You should adhere to accepted norms regarding referencing, paraphrasing, and plagiarism. See the University guides in the FinalReportDemo folder.
- Use good grammar, and make sure that you understand the proper uses of punctuation symbols such as: comma, semi-colon, colon, and apostrophe. Avoid overly long sentences: 30 words is generally too long.
- Avoid unnecessary repetition.
- The standard academic style (to which you should conform) is past tense, third-person.
- Use passive forms when possible (e.g., 'the software was tested' not 'I tested the software').
- Try to be formal in style; avoid slang and colloquial language.
- Avoid being conversational in style. Avoid personal pronouns such as 'I', 'me', 'my', 'we', 'you', 'he', 'she' and try to avoid other personalisations such as 'the author', and 'the student'. Do use phrases such as 'this project', 'the program', 'the respondents' (and do use 'it' or 'they' to avoid excessive repetitions of these). Refer to yourself ('the author' – possibly followed by 'I' or 'my') only when you are clearly stating your own personal opinion.

Final Deliverables

The report word limit is *strictly* 10 000 words. Your reference list, bibliography, and appendices are not included in the word count. We suggest that your report should be no less than 8000 words.

You should also upload an electronic (pdf) copy of your report to the SPMS. Please upload a single pdf file rather than a zip of multiple files.

Defense

- In the absence of a demonstration / defense, you will receive a mark of ZERO for your project.
- The demonstration/presentation will be starting from *TBA*. The schedule and venue will be informed to you via DLE.
- The demonstration can make a significant impression (either good or bad!) on the markers
- Please note that you will usually not be allowed to install any further software on NSBM machines.
- If you need to reschedule your demonstration. We expect to receive requests for demonstration rescheduling only under exceptional circumstances. Assuming that your supervisor agrees that you have a valid reason for requesting a rescheduling, then markers will do their best to accommodate. You should note however that a rescheduling may simply not be possible.

- The demonstration itself - You should plan your demonstration carefully. It will normally last 40 minutes; a suggested breakdown is as follows:

    - Introduction (7 minutes): don't assume that your markers have read your report. Start by telling them about your project: the client, the motivation, research, what you have produced (and why); an opportunity to set out your stall (what you want to be assessed on) and bring out the credit-worthy aspects.
    - Product demonstration (23 minutes): show the markers your product. When you do this, you need to consider that we are assessing a Computing final stage project – we want to know about both the user-oriented features and the underlying academic /computer science/software engineering issues. You could also tell the markers about the development process.
    - Q&A (10 minutes)
      We do not normally require you to provide a formal presentation (say using PowerPoint).

- If in doubt about any of the above (perhaps you have a somewhat non-standard project) then please discuss with your supervisor.
- Feedback Following the demonstration, the markers will try to provide you with some brief feedback; they will not provide you with marks.