Notes on Doing a Project

3rd year / 4th year

Susan Stepney

Department of Computer Science
University of York

Sample Project Table of Contents

1. Introduction
   - scope and context of entire project
2. Literature review
3. Overview
   - scope and context of your work
   - requirements / null hypothesis
4. Design
   - design of the artefact / experiment
5. Implementation
   - implementation of the artefact / experiment
6. Evaluation
   - how well your artefact meets the requirements / whether you have rejected your null hypothesis
   - how well your approach worked
7. Conclusions and future work
8. Bibliography

Effort and plan

How much effort?

- how many hours for the project?
  - 40 / 50 credits
  - 10 credits = 100 hours
  - 400 / 500 hours, over two 10-week terms
    - ~ 20 / 25 hours a week, or 4 / 5 hours *a day*, on average
- how many pages?
  - word limit = 35,000 words and 70 pages
  - typing at 30 wpm, that's ~ 20 hours, just typing flat out!
    - plus figures, checking, editing, polishing, binding, …
    - better to do it incrementally
- you will need a plan and timetable!
  - plan the work, plan the report

Planning

- decide
  - what has to be done
  - how long it will take
- schedule some milestones/checkpoints
  - but do not overplan
- when you deviate from plan
  - "a plan is there to tell you when something is wrong"
  - work out what is *going* wrong, and replan
    - discover this sooner rather than later
- schedule slippage *cannot* be recovered
  - descope and replan, don't "work faster"

Project marking

- remember – it is *only* the written project that is marked
  - *not* the implemented artefact or experiment
- it doesn't matter how good your design/code is
  - if it is poorly written up, you will *not* get good marks
    - plan to have *at least* one draft before the final version
- take this into account in your plan
  - the most common cause of poor marks is inadequate write-up
    - don't leave it all until the end -- write as you go

Literature review and Bibliography

Literature Review … what to read

- gather the material you need
  - search the Web / citation indexes to find references
    - CiteSeer : http://citeseer.nj.nec.com/
  - follow up relevant references in papers you read
- make working notes on each paper/book you read
  - a paragraph of relevant points will make writing the review much easier!
- don't worry if you don't understand it all at first !
  - it can take a while to get "into" a subject
  - it *will* start making sense
  - when it does, go back and reread the earlier papers
    - see how much more sense they make now

Literature Review … what to write

- **critically review** the state of the art of the subject
  - tell a coherent story, in terms of other people's work
  - justify your critique
    - don't just say "but I don't agree"
  - use a style, for example, like an introductory textbook
    - but with references
    - read some review papers, too, to get a feel for the style
  - the review should justify the rest of your work
    - show that it's not been done before
    - show that it fills an important/interesting gap
- **reference** the literature
  - to demonstrate your knowledge of the state of the art
  - to bring together other people's work into a story

Referencing style

- tag styles
  - number : [42]
    - by order of occurrence in the text
    - by alphabetical order in the bibliography
    - *not* by the order in which you wrote them down !
  - initials and year
    - [Smi87], [S&J88b], [CBS94]
  - names and year
    - [Smith 1987], [Smith & Jones 1988b], [Cobley *et al.* 1994]
- using the tags
  - "[42] describes the results of Smith & Jones, which show X is better than Y"
  - "the results of [Smith & Jones 1988b] show X is better than Y"
  - "X is better than Y [S&J88b, chapter 8]"

An old joke, adapted

As they arrive in York, a Software Engineer, a Computer Scientist and a Mathematician see a black swan on the lake.

The Software Engineer thinks all York swans are black.

The Computer Scientist decides that some York swans are black.

The Mathematician concludes that in York there is a lake containing at least one swan, of which at least one side is black.

Referencing in the text

✗ [SwEng98] claims the blackness of all York swans, whereas [CompSci03] notes that some York swans are black. On the other hand, [Math57] carefully establishes that in York there is a lake containing at least one swan, of which at least one side is black.

✓ [SwEng98] claims the blackness of all York swans, whereas [CompSci03] notes that "some York swans are black". On the other hand, [Math57] carefully establishes that

> "in York there is a lake containing at least one swan, of which at least one side is black."

Bibliography

- Bibliography = the list of references at the end

- include sufficient information that someone else could find the work
  - author
  - title
  - source
    - if off the Web: URL (but don't have too many of these)
    - if a book: publisher
    - if a paper: book/journal (including report/volume number)
      - and then source of book…
  - year

Bibliography : examples

[Wil90]

J. M. Williams. *Style: towards clarity and grace.* University of Chicago Press, 1990.

[Ste03]

S. Stepney. Review of J. M. Williams' *Style.* April 2003.
http://www-users.cs.york.ac.uk/~susan/bib/nf/w/jsphmwll.htm
(accessed 13 October 2003)

[Med63]

P. B. Medawar. "Is the scientific paper a fraud?" *The Listener*, **70**, 377–378, Sept 1963.

[P&C93]

D. L. Parnas, P. C. Clements, "A Rational Design Process: how and why to fake it". *IEEE Trans. Sw. Eng.*, **19**(2), 251–257, Feb 1993.
Available from:
http://www.cobolreport.com/columnists/parnas&clements/09152003.asp

Packages

- LaTeX
  - use BibTeX
    - ensures consistency
    - can flip between styles
    - beware of case changing!
- MS-Word
  - use a *consistent* style
    - check out how the journals do it
  - numbers are hard to keep up-to-date
    - Insert > Cross-reference…
      - tricky to use

The engineering lifecycle for “experimental“ projects

BCS/IEE accreditation requirement

- Components of engineering lifecycle project
  - Requirements
  - Design
  - Implementation
  - Evaluation
- this can be of
  - software
  - hardware
  - experiment
  - …

The null hypothesis

- the “question” is phrased as a **null hypothesis**, H_0
- usually, H_0 is a statement of the status quo
  - “the change has no effect”
  - “the new design process is no different from the old”
  - “the new parameter values give the same results as before”
- H_0 assumed true unless data indicate otherwise
  - the experiment is trying to reject the null hypothesis
  - can reject, but cannot prove, an hypothesis
    - evidence that supports the hypothesis may just be a fluke
    - “all swans are black”
      - one white swan suffices to reject
      - no number of black swans can prove -- the next swan could still be white

Can reindeer fly?

- you believe reindeer can fly
- null hypothesis: "reindeer can*not* fly"
- experimental design : to throw reindeer off the roof
- implementation : they all *go* splat on the ground
- evaluation : null hypothesis not rejected
  - this does not *prove* reindeer cannot fly: what you have shown is that
    - "from this roof, on this day, under these weather conditions, these particular reindeer either could not, or chose not to, fly"

— [Christmas Guardian, 1980s, paraphrased]

- it is possible, in principle, to reject the null hypothesis
  - by exhibiting a flying reindeer !

Experiments as Lifecycle Projects

- **Requirement** : statement of the aims of the experiment
  - to test the null hypothesis: "new coding style has no effect on readability"
- **Design** : of a controlled experiment to perform the test
  - you have to measure readability using both new *and* original style, to have something to compare, and to measure the effect
  - *the control* : the experiment on the original style
- **Implementation** : carrying out the experiment
- **Evaluation** :
  - do the results reject the null hypothesis?
    - if no, iterate with further design/implementation/evaluation
    - if yes, iterate with new hypothesis
  - *meta-evaluation* : how well did your approach work?

Two sets of data

- hypothesis-generating data
  - used to suggest the null hypothesis
    - you have data suggesting a coding style improves readability
      - null hypothesis: “new coding style has no effect on readability”
      - cannot use original data to reject the null hypothesis, because it was used to suggest it! (“all swans are black”)
- further *independent* test data
  - used to test the null hypothesis
    - measure the readability of new samples of old and new style code
      - no significant effect on readability : null hypothesis not rejected
      - improvement in readability : null hypothesis rejected
      - *decrease* in readability : null hypothesis also rejected !

Double blind experiment

- testing a subjective effect, like "readability"
    - control group : original style
    - experimental group : new style
  - subjective : knowledge of which style can influence answers
- so, perform "double blind"
  - *neither* experimenter *nor* subject know who is in the control and who in the experimental group
    - randomised allocation
    - look at membership after results have been *collected*
      - are the two groups significantly different?
    - look at membership after results have been *analysed*
      - are there two populations that correspond to the two groups?

in conclusion …

Remember …

- The work
    - it is *only* the written project that is marked
    - there is a lot of your effort involved
  - you need to plan and monitor your progress
- Use your supervisor
  - guidance on the subject area
  - guidance on the project plan
  - guidance on the report structure
    - comment on drafts
    - but only useful if provided early enough for me to read
      - and for you to do anything with my comments !