COM3600
Undergraduate Dissertation Project 2016/2705

Session #1

Doing an Undergraduate project

Dr Amanda Sharkey
Department of Computer Science
University of Sheffield

a.sharkey@sheffield.ac.uk

Aims of this session

- To review the course contents
- To review the schedule for the Dissertation project
- To explain the key steps in planning your project
- To review common mistakes in scientific writing
- To explain how to write a literature review
- To explain how your project will be assessed

Overview of this course

Lecture sessions

Week	Topic
1	Doing an Undergraduate Project
2	Presentations and Posters (Dr Ramsay Taylor)
3	Assessing and managing risk (Dr Ramsay Taylor)
4	Legal issues 1: Intellectual property and relevant legislation (Prof Guy Brown)
5	Legal issues 2: Contracts, confidence and liability (Prof Guy Brown)
6	Ethical and professional issues (Dr Amanda Sharkey)

You should also be meeting your project supervisor on a regular basis; arrange meetings now!

Assessment

- Description stage: Semester 1, Week 2, Friday 7th October 2016
- Survey and Analysis Stage: Semester 1, Week 11, Monday 5th December 2016
- Final Dissertation: Semester 2, Week 10, Wednesday 3rd May 2017
- Poster Session: Semester 2, Week 11, Wednesday 10th May 2017

Feedback

- You will receive feedback as follows:
  - Description Stage: written and verbal feedback from your supervisor and second marker
  - Survey and Analysis Stage: written and verbal feedback from your supervisor and second marker and an indicative mark
  - Final Dissertation: a mark, and written feedback
  - Poster Session: verbal feedback at the time, and written comments (included with those on dissertation)

- Description Stage
  - Short report, no more than 4 pages
  - Typical structure:
    - Title, name, supervisor, module code, date
    - Introduction: background, clear description of project, and cite literature read to date
    - Analysis: discussion of problems to be solved and possible techniques and tools
    - Plan of action: weekly plan of work

» Survey and Analysis stage:

No longer than 25 pages

Main components: extensive literature survey (or similar technology/ mathematical survey for some types of project)

A clear presentation of what the project is aiming to achieve, and a description of work done so far.

Feedback: from supervisor and second marker, which will help you to prepare your final dissertation.

You can reuse what is written for this stage in the final dissertation

Failure to submit a satisfactory report will result in disciplinary action

- Aim: to make sure project is feasible
  - Is it clearly defined?
  - Is it feasible?
  - Does the student know what they are doing?
  - Have potential problems and possible processes/tools/techniques been identified?
  - Is the plan of action realistic?

Planning your project

Why plan at all?

Why plan at all?

- Ensure that end point of project is feasible
- Enable communication between you, supervisor and (if applicable) project client
- Highlight problems early
- Help manage risk
- You can’t control a project that hasn’t been planned!

Key steps in planning

- Identify project scope and objectives

What are the objectives of the project?

Who are the stakeholders?

How will you communicate with the parties involved?

Key steps in planning

- Identify project scope and objectives
- Identify project infrastructure

Will your project have to fit with an existing system?

Is there particular hardware or methods that must be used?

Is a client involved who will be required to sign off your work?

Key steps in planning

- Identify project scope and objectives
- Identify project infrastructure
- Analyse project characteristics

Is your project objective-driven or product-driven?

What are the high-level risks?

What software development approach is suitable?

Key steps in planning

- Identify project scope and objectives
- Identify project infrastructure
- Analyse project characteristics
- Identify project products and activities

Identify the project products (deliverables)

Identify the activities needed to get you there

How will you judge the quality of the products?

Key steps in planning

- Identify project scope and objectives
- Identify project infrastructure
- Analyse project characteristics
- Identify project products and activities
- Estimate effort for each activity

Estimate amount of time (effort) required for each activity

Supervisor can guide you on this

Should long activities be broken down?

Project feasible in timescale?

Key steps in planning

- Identify project scope and objectives
- Identify project infrastructure
- Analyse project characteristics
- Identify project products and activities
- Estimate effort for each activity
- Identify activity risks

See later session on risk analysis and management

Risk reduction and contingency planning

Key steps in planning

- Identify project scope and objectives
- Identify project infrastructure
- Analyse project characteristics
- Identify project products and activities
- Estimate effort for each activity
- Identify activity risks
- Allocate resources

Who is going to do what?

Do you need input from your supervisor, or from another source?

Key steps in planning

- Identify project scope and objectives
- Identify project infrastructure
- Analyse project characteristics
- Identify project products and activities
- Estimate effort for each activity
- Identify activity risks
- Allocate resources
- Review and share plan

Share plan with supervisor

How will you know when each activity has been completed?

How will you report a completed activity?

Key steps in planning

- Identify project scope and objectives
- Identify project infrastructure
- Analyse project characteristics
- Identify project products and activities
- Estimate effort for each activity
- Identify activity risks
- Allocate resources
- Review and share plan
- Execute plan, with regular reviews

When project is underway, some stages may require more detailed planning

Regularly review plan with supervisor

Long-term provisional plans should be firmed up

Activities

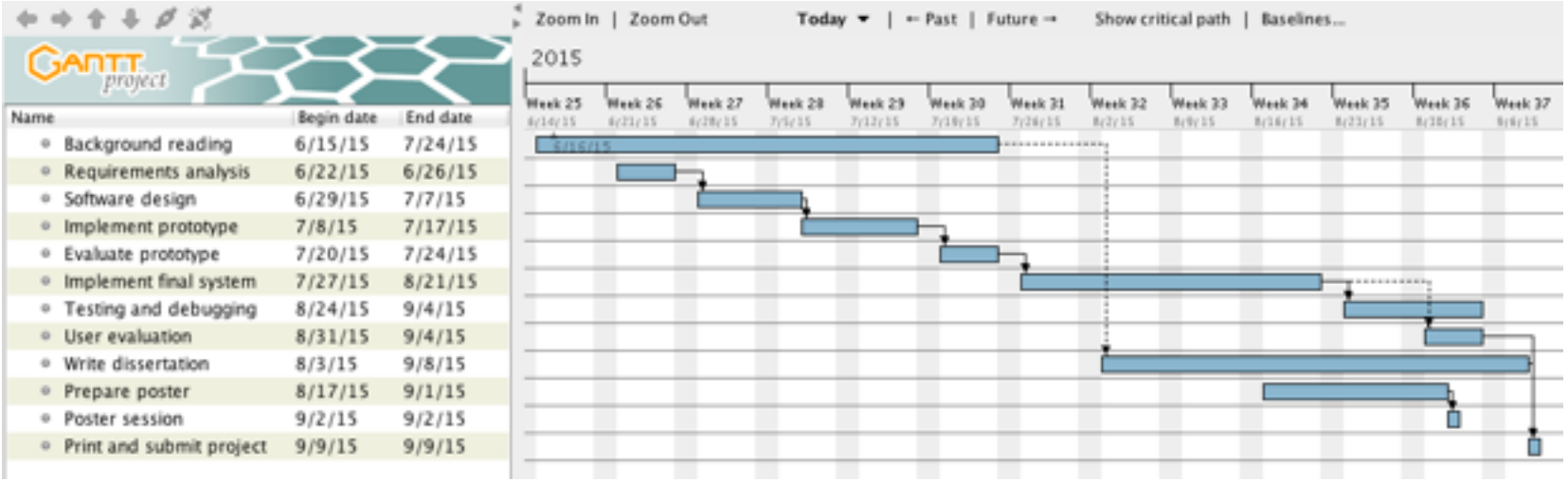
- A key aspect of planning is identifying the **activities** that you must do to complete the project
- Not sufficient just to list them!
- For each activity, define the following:
  - **Pre-requisites**: what must be done before the activity can start?
  - **Dependent activities**: which activities need this one to be completed first?
  - **Estimated time/effort**: could be a range of values
  - **Quality checks**. How are you going to verify/validate the product of the activity?

Critical path and float

- The **critical path** is the chain of sequential activities that determines the minimum time required for the project.
- Passes through the activities with least **float**.
  - The float is the difference between the earliest start (or end) date of an activity and its latest start (or end) date
  - Work out how long the start or completion of an activity may be delayed without affecting the end date of project
- Any delay on the critical path will delay the project.
- Can be done using a method such as PERT or CPM
- Software tools (e.g. Microsoft Project) can help you

Gantt chart

- The usual way of showing start and finish dates of planned activities (and progress against them)
- Time vs. work breakdown structure
- Include information about predecessors
- Free software tools (e.g., www.ganttproject.biz)



Name	Begin date	End date
Background reading	6/15/15	7/24/15
Requirements analysis	6/22/15	6/26/15
Software design	6/29/15	7/7/15
Implement prototype	7/8/15	7/17/15
Evaluate prototype	7/20/15	7/24/15
Implement final system	7/27/15	8/21/15
Testing and debugging	8/24/15	9/4/15
User evaluation	8/31/15	9/4/15
Write dissertation	8/3/15	9/8/15
Prepare poster	8/17/15	9/1/15
Poster session	9/2/15	9/2/15
Print and submit project	9/9/15	9/9/15

Writing your project plan

- I suggest that your project plan consists of:
  - Gantt chart
  - Under the chart, a description of each activity in the work breakdown structure (with details of prerequisites, dependencies etc)
  - Some commentary on the critical activities; those with least float
- A full critical path analysis is not required; but if you do it, the critical path can be shown on the Gantt chart.

Writing your report

Structure of a typical UG dissertation

Title page
Signed declaration
Abstract
Acknowledgments
Contents
1. Introduction
2. Literature review
3. Requirements and analysis
4. Design
5. Implementation and testing
6. Results and discussion
7. Conclusions
References
Appendices

- This is a **typical** structure; it may not suit all projects (consult with your supervisor)
- Maximum 60 pages in 11pt or 12pt
- Appendices are not included in page count, and are not assessed

Style

- Your dissertation should be written in **scientific English**.
- Avoid jargon; write as clearly as possible
- Keep your intended audience in mind; a competent computer scientist who is not necessarily a specialist in the topic of your project
- Writing should be formal, not colloquial. You should not:
  - Use contractions, e.g. “don’t”
  - Use information words, e.g. “lots”
  - Use emotive words, e.g. “terrible”, “fantastic”, “awful”
  - Write in the first person, i.e. avoid the use of “I”

Other writing pitfalls

- Flat and uninspiring introductions that don't make the motivation for the study clear
- Poor paragraphing
- Ambiguity
- Poor sentence structure
- Double negatives

Some examples are from Zobel (2004) *Writing for computer science*, Springer.

What's wrong with this? (1/6)

- *I constructed the system*
- Writing in first person. This is preferred:
- *The system was constructed*

What's wrong with this? (2/6)

- *Use of digital libraries is increasingly common.*
- A flat and uninspiring introductory sentence which actually tells the reader very little. This is better:
- *Digital libraries provide fast access to large numbers of documents.*

What's wrong with this? (3/6)

The rise of the internet has contributed to the emergence of an interconnected world, it's now taken for granted that individuals who are physically far apart can communicate with each other using suitable hardware and video conferencing software which transmit audio and video in real time, whilst these technologies do facilitate communication over long distances it is always a compromise to actually "being there", presence, the benefits of being present is weighed against costs such as travel, when the costs of being present outweigh the benefits alternatives like video conferencing are used, in some scenarios whilst the cost of being present may be great, cheaper alternatives like video conferencing aren't an option due to the need to interact with a remote environment, but what if a person's presence could be transmitted and embodied in an avatar which is actually physically present in the remote location?

What's wrong with this? (4/6)

- *The compiler did not accept the program because it contained errors.*
- Ambiguity – are the errors in the program or the compiler? This is much better:
- *The program did not compile because it contained errors.*

What's wrong with this? (5/6)

- *In the first stage, the backtracking tokeniser with a two-element retry buffer, including illegal adjacencies as well as unrecognised tokens, are stored on an error stack for collation into a complete report.*
- Sentence structure is a problem here; nested sentences are hard to understand. This is much better:
- *The first stage is the backtracking tokeniser with a two-element retry buffer. In this stage, possible errors include illegal adjacencies as well as unrecognised tokens; when detected, errors are stored on a stack for collation into a complete report.*

What's wrong with this? (6/6)

- *There do not seem to be any reasons not to adopt the new approach.*
- Double negatives are hard for the reader to understand, and potentially ambiguous. The intention here was meant to be to praise the new approach, but it actually reads like condemnation. This is much better:
- *The new approach is at least as good as the old, and should be adopted.*

Plagiarism

- We will look at ethics later in the course, but it is worth noting plagiarism as a particular problem at this stage.
- Plagiarism is re-use of material in one paper that has appeared in another, without appropriate acknowledgment.
- Applies to images as well as text.
- *If you cut and paste when taking notes for your dissertation, highlight the text, or change it to italics, to remind you that these are not your words.*
- When writing do not use other text as a guide, use quotation marks for borrowed text, and always cite your references.
- All dissertations are run through the TurnItIn system.

- Citing references - to avoid problems with unfair means
- Also citing references to back up your statements strengthens them
- It shows that you have read and are aware of related research in the area

Referencing styles

Harvard style:

Some early work by Brooks (1975) suggests that throwing personnel at a problem is counter-productive.

Brooks, F.P. (1975) The Mythical Man Month: Essays on Software Engineering. New York: Addison-Wesley.

IEEE style:

Early work in this area [1] suggests that throwing personnel at a problem is counter-productive.

[1] Brooks, F.P. (1975) The Mythical Man Month: Essays on Software Engineering. New York: Addison-Wesley.

Literature review

- What is the **purpose** of the literature review?

Literature review

- What is the **purpose** of the literature review?
  - Establishes a framework for your project topic
  - Introduces key terms, definitions and terminology
  - Identifies previous studies, models, case studies relevant to your project
  - Defines and limits the area for your own project
  - Shows that you are aware of the background to your project
    - what has been done before
    - does your project do something novel?
  - Demonstrates your ability to organise, and understand, a body of work relating to your project

Literature review:

- Start with a general overview, and gradually focus in on your particular project - setting it in context
- Not just a summary – your review must be **critical**:
  - e.g. Criticise aspects of methodology
  - e.g. Highlight gaps in knowledge that your project might fill
- It should contain both **analysis** and **synthesis**:
  - Compare and contrast different views on an issue
  - Group studies that draw similar conclusions
  - Note areas in which studies are in disagreement
- Final part of review should be **focused**, and **linked** to your project goals:
  - Show how your project relates to previous studies and the literature in general
  - Identify key concepts and methods that you will take forward

Evaluation

- Often overlooked until the end of a project, and then it is too late!
- How are you going to evaluate your work?

Evaluation

- Often overlooked until the end of a project, and then it is too late!
- How are you going to evaluate your work?
  - For **software development** projects, both functional testing and user-acceptance testing are appropriate
  - For **experimental** projects:
    - compare against existing techniques on common data set if possible
    - OR use descriptive and inferential statistics to evaluate the results and assess their significance
  - For **theoretical** projects, compare against relative power/ expressiveness of competing approaches
  - The ability to objectively self-assess your work is key; including results, products and processes.

How will your dissertation be assessed?

Assessment

Quality of products	Results of the work (software, hardware, models, theorems etc). Takes into account difficulty of task and success achieved.
Quality of processes	Analysis, design, testing and relevant legal, social and ethical issues. Use of literature, tools and methodologies.
Amount of work completed	Background learning and final products. Judged in relation to what could reasonably be expected, given time available and level of difficulty.
Quality of evaluation	Your own evaluation of the project; how well objectives were met, critical assessment of your processes and results, suggestions for future work.
Presentation	Readability, quality of writing, notation, diagrams, layout, referencing etc.
Poster session	Poster itself, software demo, question answering

Summary

Summary

- Planning is important, even for a short project
- A Gantt chart is not enough
  - Think about dependencies between activities
  - How will you assess the quality of the product from an activity?
  - Identify critical activities which could lead to delays
- The best way to understand how to write in scientific English is to read technical papers
- Keep in mind the purpose of a literature review as you write it

Bibliography

- J. Zobel (2004) *Writing for computer science*. Springer.
- B. Hughes and M. Cotterell (1999) *Software project management*. McGraw Hill.