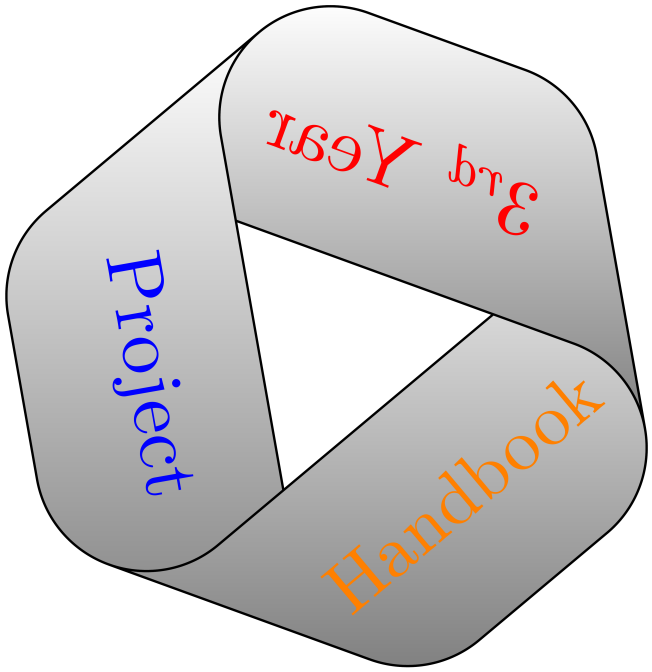
3rd Year Project Handbook

2023/2024

Department of Computer Science
Adran Cyfrifiadureg

February 20, 2024



Contents

1 Introduction

The 3rd Year Project, or simply, the *project*, is a major part of your Computer Science education. It constitutes a quarter of the year and will often be the largest undertaking you have made until now.

The project will allow you to try out many of the things that you have learnt over the last years. You will also, undoubtedly, need to learn many new things while completing the project. This will hopefully be a both fun and rewarding learning experience.

The project allows you to delve deeper into some area(s) of interest. But there are also many common areas from your studies that will naturally tie in to the project. For example, planning, software engineering practices, presentation skills, and so on. Notice that these are important transferrable skills for your future working life, in whatever role you find yourself. You will be presented with new challenges, partly due to the scale of the project, but also because you are expected to lead your project. This is often the first time students are asked to assume this challenging role.

A "final year project" has been a cornerstone in the final year of most Computer Science programmes for a long time. Students are taught many things in different modules, but to be able to effectively work in the Computer Science field a practitioner must be able to connect many different pieces of knowledge. It is very difficult to do this within the framework of a regular module, but within a project this ability can be shown. Thus, a final year project is fundamental for BCS accreditation, and it is also an important merit for employability.

2 3rd Year Project Timeline

All 3rd Year Projects follow the same timeline regardless of the programme. There are currently four deliverables for the project. These are:

Project Specification and Planning Document	30/10/2023
Gregynog Presentation	26/11 – 2/12/2023
Project Dissertation	26/4/2024
Project Fair Poster Presentation	7–8 May 2024

Project selection starts already at the end of Year 2. Final project allocation happens in September before start of Year 3. In preparation for the project students may use the summer for background reading and research of topic ideas.

There will be four lectures given during the year that targets the four deliverables listed above.

Lecture on Specification and Planning Document	5 Oct 2023
Lecture on Gregynog Presentation	2 Nov 2023
Lecture on Dissertation	29 February 2024
Lecture on Project Fair	18 April 2024

Students will also complete career development activities as part of one of the project modules. We will not cover that in this document. See Canvas for information about the career development component.

3 3rd Year Project Modules

The project consists of 2 modules that together cover the learning outcomes:

- Project Specification and Development module, and
- Project Dissertation module.

Both modules are worth 15 credits; making the project worth 30 credits in total. Depending on the strand and on the kind of project chosen, there will be different module codes, but they serve the same purpose.

Both modules run over both terms.

3.1 Project Specification and Development Module

This module consists of 4 components:

Specification and Planning Document	35%
Gregynog Presentation	25%
Project Fair	30%
Career Development	10%

This module gives some structure to the development of the project by isolating some pieces that prompt the progression of the project.

The module also practises important transferrable skills such as presentations and document writing.

3.2 Project Dissertation Module

This module consists solely of the final written document presenting the product/results achieved during the project.

3.3 Project Flavours

3rd year projects come in different "flavours". These have different module codes as shown in Table 1.

Table 1: Project Modules

	Development module	**Dissertation module**
Software Engineering	CSP301	CSP300
Computer Science	CSP354	CSP344
Non-technical	CSP354	CSP302

The time-line presented in Section 2 applies to all flavours, i.e., all students will submit a Specification and Planning Document and a Dissertation, and all students will give presentations in Gregynog and at the Project Demonstration Fair. The same deadlines apply to everyone.

Projects on the Software Engineering strand are expected to produce a major piece of software. Projects on the Computer Science strand have a larger range, from major pieces of software to essentially no software.

3.3.1 Technical vs Non-technical project

BCS accreditation for our programmes hinges on the generation of an "artefact" covering "at least part of the implementation lifecycle". For example, the artefact could be program code, a formal specification, verification, or a test suite. Examples of projects that are not acceptable for BCS accreditation include pure literature reviews and user/market surveys.

Students on the CS strand can opt to do projects that do not meet the BCS accreditation requirements. These are known as **non-technical** projects within the department. They still need to have clear Computer Science content. Some areas within CS commonly have non-technical projects, for example, History of Computing. The vast majority of projects in the department are **technical** projects.

Non-technical projects mean the overall degree is not suitable for BCS accreditation, which has little effect other than students will not get automatic qualification for BCS membership. The degree is still a full Computer Science award, equivalent in every other aspect. Most employers are uninterested in BCS membership.

There is no direct selection of the non-technical project module (CSP302) by students. This will be agreed between the student and the supervisor shortly before or after the Specification and Planning Document is due. A module change will then be made to reflect the nature of the project.

3.3.2 Difference between project flavours

The main difference between the projects is the nature of the project management and what methodologies are used. When major pieces of software are to be produced, the main concern is software engineering. Hence, the focus would be on choosing an adequate software development model and applying it to the project at hand. For more research oriented projects, where the amount of code can be less, other methodologies may be more suitable. What methodologies are used often depend on the research area. Using appropriate methodologies is essential for a successful project and should be chosen carefully.

3.4 Project Information Online

There are two online sites that will contain project information: Canvas and CS Project Manager.

Canvas contains all the information about the module and this is also where any announcements will be made. The assignment submissions are also there.

CS Project Manager takes care of the project allocation. This is also where marking takes place, so this is where the marking schemes, preliminary marks, and feedback can be found.

4 Supervision

Each project will be assigned a supervisor. The role of the supervisor is to guide the student through the project. However, students are expected to lead the project.

The department commits to three hours per term of supervision. The format for supervision meetings needs to be agreed by the student and the supervisor. Some supervisors run group meetings, while others have shorter individual meetings. Learning to manage the relationship with your supervisor is an important transferrable skill.

Students are also assigned a second marker. They, together with the supervisor, will usually mark the deliverables of the project, except the Gregynog Presentation. The second marker does **not** normally take part in supervising the project.

Concerns regarding supervision or more generally any issues regarding the processes in place to manage third year projects can be reported to the Year Head, Jens Blanck, or Programme Director, Liam O'Reilly.

5 Ethics

Any research taking place within the university shall be ethical as required by university policy. As projects often contain some research elements, they shall go through the faculty system for Research Ethics Applications. Some further information is available at this FAQ.

There are two main categories of projects that are exempt from this process:

- Literature review and meta-analysis that make use of data or information already in the public domain.
- Student practicals and demonstrations of competence that do not involve the collection of data from/about living organisms or human subjects.

A project shall only proceed if necessary approval has been gained or the project is exempt by the above. Note that user studies do collect data from humans, and therefore need ethics approval.

6 Academic Integrity and Academic Misconduct

By submitting coursework, electronically and/or hardcopy, you state that you fully understand and are complying with the University's policy on Academic Integrity and Academic Misconduct. The policy can be found at `https://myuni.swansea.ac.uk/academic-life/academic-misconduct`. The consequences of committing academic misconduct can be extremely serious and may have a profound effect on your results and/or further progression. The penalties range from a written reprimand to cancellation of all of your marks and withdrawal from the University.

6.1 On-line repositories and collusion

On-line repositories, such as GitHub, Bitbucket and GitLab, are useful to store project related data to track changes and reduce chances of data loss. Unfortunately, in the context of Academic Misconduct it is possible that other site users can use data you make publicly available. This may inadvertently allow other students to base their submission on your work. This may cause you to be dragged into a potential collusion case. Therefore, we recommend that you restrict access to your data wherever possible by setting the repositories to private. It is your responsibility to protect your data. After your project is marked, you can make it public to showcase your work, e.g., for employability purposes.

7 Deliverables

7.1 Project Specification and Planning Document

This written document is due 30/10/2023. It is worth 35% of the development module.

The document shall not exceed 7500 words (approximately 15 pages) of core content (introduction to summary) and shall demonstrate that the student

- has clear aims/objectives for the project,
- can provide good motivation,
- can place the project in a wider context,
- has done sufficient background research,
- has good understanding of the tools and techniques that will be used,
- has chosen an appropriate software development model, or other methodology,
- has a realistic work schedule fitting the chosen development model/methodology,

- has identified potential risks and their mitigations, and
- is able to write a document in a style suitable for professional and academic settings.

The specification and planning document is not completely binding. We recognise that many student projects evolve during the course, and that the plans may have to be adjusted. However, this is not an excuse to only do superficial planning. The plan should be the best possible at the time of submission.

The document shall be submitted as a pdf file through Canvas. Students may be contacted to provide a further copy by email.

The document will be assessed by the supervisor and the mark will be moderated by the second marker. Marking scheme is available at CS Project Manager.

7.2 Gregynog Presentation

This is a conference style presentation made to a small group of peers. It is worth 25% of the development module.

Students are divided into cohorts and will go to Gregynog for their presentations during 26/11 – 2/12/2023.

The presentations slots are usually 10–15 minutes long, followed by a short time for the student to answer questions. The students shall show that they have:

- an exciting topic for their project with motivation and aims/objectives,
- good knowledge about the tools and techniques needed to complete the project, and
- made reasonable initial progress on their project.

Students may be asked to submit their presentation slides for documentation purposes.

7.3 Career Development

There will be some short assessment, usually in the form of Canvas quizzes. The component is worth 10% of the development module.

7.4 Dissertation

This written document is due 26/4/2024 and is worth 100% of the dissertation module.

We would like to emphasize that this time of the year is a very busy time for students. We urge students to start early and plan sufficient time to complete this document. In a well-managed project this should, of course, not be a problem.

It is usually very beneficial to prepare an outline and discuss it with the supervisor before the Easter break. Having a fairly settled outline makes the writing much easier. Additionally, easy improvements that will significantly strengthen the product are often found during such a discussion.

The dissertation is a comprehensive document detailing the delivered product/results, the process leading up to this, and a reflective account on the product/results and the development of it.

The document shall not exceed 15000 words (approximately 30 pages) of core content (introduction to summary). The document shall:

- recap the aims/objectives and show to what extent they have been met,
- place the product/result into context, e.g., how it compares to other efforts in the same direction,
- give a detailed account of how the product was built and how results were obtained, and
- reflect on the achievements and processes of the project.

It is acknowledged that project aims and objectives may have changed during the course of the project which is natural and not a problem. However, these changes should be described and explained.

The document shall be submitted as a pdf file through Canvas. Students may be contacted to provide a further copy by email.

The document will be assessed by the supervisor and the mark will be moderated by the second marker. Marking scheme is available at CS Project Manager.

7.5 Project Fair

This is an interactive poster style presentation made to academic staff, students and visitors from industry. It is worth 30% of the development module.

Students will be assigned a stand where a large poster can be displayed. There is also room to place a laptop on the stand (which will be critical for many projects). Students shall:

- attract the attention of visitors at the Fair,
- be able to succinctly convey
  - what the project is about,
  - what methods were used,
  - how the problem was solved, and
  - what the results are;
- be prepared for various length interactions, from 2 to 8 minutes, and
- have a running demo, which the audience can interact with, if at all feasible.

8 Project Management

8.1 Software Engineering

Most projects will develop some software. Students are expected to follow good Software Engineering practices as taught in previous modules. For more detail, see, for example, these recommended books from previous modules.

- I Sommerville: *Software Engineering*, 10th ed., Pearson, 2015.
- R Pressman and B Maxim, *Software Engineering: A Practitioner's Approach*, 9th ed., McGraw-Hill, 2019.

8.1.1 Software development model

For any software development, students shall choose a software development model (sometimes called software development lifecycle (SDLC) model). The choice shall be explicitly made and rationalised. This is of even higher importance for larger or more complex software. The chosen model can be adapted to better fit the constraints of the project.

8.1.2 Code quality

Produced code is expected to be of good quality, i.e., well-formatted code that is appropriately commented.

8.1.3 Software testing

Depending on the software generated there may be several levels of testing that are relevant. For example, unit testing, integration testing, system testing and acceptance testing. There are many more testing types for more specific scenarios.

Students shall apply a suitable testing suite to the generated software and document the test results. Formal/automated testing is preferred to manual ad hoc testing.

8.2 Aims and Objectives

8.2.1 Aim

An *aim* is what is hoped to be achieved. Usually only one, or possibly two, are formulated. It can be fairly broad. It should be challenging but achievable. The aim is often phrased with the infinitive.

> The aim is ... *to solve, to provide, to generate* ...

8.2.2 Objectives

Objectives are what will be done to achieve the aim. Usually more than one objective is stated per aim. Objectives are the concrete tasks that will be performed. They shall be realistic. They shall also be precise and it shall be possible to determine if they have been met (this is known as being measurable). They are usually presented as a numbered list. Usually phrased with active sentences.

1. We will ... *build, construct, design, test, verify* ...
2.
3.

8.2.3 Common problems

- Too many aims. *It pays off to find one focus for the project.*
- Aims and objectives are just reformulations of each other.
- Aims and objectives are not precise enough. *Without a clear picture of what is to be done, good progress is unlikely.*
- The objectives do not support the aim.
- Non-achievable aims or objectives within the time-frame.

8.3 Milestones and Deliverables

The project will have a number of deliverables. Some are common to all projects, such as the project module assessments, but others are unique to individual projects. The deliverables can be many things, such as: programs, data sets, specifications, documents and presentations.

The project may also have a number of milestones. Milestones are key stages in the development where the project moves into a new phase. They often coincide with deliverables, but they can also be internal or external signals that the project can proceed. An example of an external signal would be achieving approval from an ethics board or similar. An example of an internal signal would be completion of initial data collection.

Careful identification and planning of milestones and deliverables, and tracking of their completion, is an important tool to keep projects on track.

8.4 Keep Notes

Keep notes during the project, perhaps in the form of a project diary. Good notes will help to keep a project on track and will also help with writing the final dissertation. Suggested information to collect includes:

- What has been done on the project. *A detailed account is needed for the final dissertation.*
- How things were done. *It is surprisingly easy to forget what commands were used to achieve the early steps.*
- What you have read. *This is needed for referencing.*
- When milestones and deliverables were achieved. *Continuously monitor the progress to ensure that project is on target.*
- Any changes made to project specification and the reason for them. *Again, this is needed for the final dissertation.*

8.5 Risks

Any project is susceptible to *risks* that may prevent it from achieving its target. An important part of project management is to assess these risks and plan for avoidance and mitigation. Students shall produce a comprehensive list of potential risks and avoidance/mitigation plans.

Consider both external and internal risks. An external risk is one that is largely out of the control of the student. Examples could be pandemics, industrial action, changes to requirements made by customer, and so on. An internal risk is one where the student has some or full control of the cause. Examples here could be poor planning, poor choice of methodology, lack of resources, and so on.

Consider both generic risks to projects, e.g., overly ambitious objectives, and risks specific to this particular project, e.g., incompatible libraries chosen.

Risks shall be assessed on both severity and likelihood. Reasonable mitigation shall be suggested. It is expected that this is tabulated for easy overview.

It might be worth pointing out, that it is risks to the *successful completion of the project* that is of relevance here. Not risks to the student, and not risks associated with the use of generated software in production environments. Although, if there are potential risks with using the software, then that may well be an issue for the ethics board, see Section 5.

8.6 Quality Assurance and Testing

Achievements of the project shall be honestly and transparently reported. This is part of the university's policy on Research Integrity. Thus, it is important to assess to what extent the achievements meet the aims and objectives set out. It is expected that suitable testing is made to support this assessment.

Depending on the type of the project, the testing may take many forms: software testing should be performed on software development projects; statistical tests may be performed on numerical data sets obtained; formal proofs may be written out for theoretical research; formal verification may be applied to formal specifications; and so on. It is up to the student to supply sufficient supporting evidence to show to what extent the aims and objectives have been met. An evaluation strategy should be created early in the project as this will provide a mechanism for judging the success of the project upon conclusion.

9 Writing Documents

It is assumed that students have learnt to write good prose in education prior to university. It is, of course, expected that assessed documents are free from spelling and grammatical errors; and that proper capitalisation and punctuation is used. We will mention some things particular to academic writing below.

9.1 Academic Writing

Academic writing has particular norms but it is not about using fancy words or convoluted sentences. In fact, the main purpose is to communicate clearly. To achieve clarity the main differences are:

- **formal language** is used as informal language often is more ambiguous,
- **precise** statements that reduce possible misinterpretations,
- **structured** documents that make it easier to locate important information and allow quick navigation, and
- **careful** selection of content to avoid swamping important information with masses of text of only marginal relevance.

9.1.1 Formal language

We summarise some common changes to normal writing.

Contractions such as **won't**, **it's** and **they're** are not used but rather written out as **will not**, **it is** and **they are**, respectively. This is for clarity, but has the added benefit of avoiding the common mixups of **it's** and **its**; and **they're** and **their**.

Most academic writing is expected to be objective. This has as consequence that passive constructions are much more common as it removes a personal pronoun and thus also the focus from the writer. For example, **I/We analysed the data** is usually written as **The data was analysed**. Another common pattern is to include the reader by using **we** instead of **I**. For example, **We can see from the table that ...**, and **We prove ...**, where we implicitly invite the reader to follow through and do the same action. In fact, the pronoun **I** is usually completely avoided in academic papers. However, in writing reflections on work done this may be inconvenient.

Do not use complex language or expressions that are not your own. Straightforward language ensures that the intended idea is accurately conveyed from the writer to the reader.

Write succinctly and avoid empty fillers in the text. The sentence "**In the end we chose the first alternative as the other alternative was deemed inferior, because ...**" is better written as "**We chose the first alternative because ...**" as there is no important information in the omitted words.

Introduce abbreviations and acronyms, if needed, on first occurrence of unabbreviated term. For example, **the Java Virtual Machine (JVM) can execute the compiled bytecode**. We can later use the acronym **JVM** directly.

9.2 Figures and Tables

Figures and tables shall be numbered and have captions that precisely states what they contain. If they are not your own then they shall have proper citations. All figures and tables should be referenced in the text so the reader knows at which point to look at the figure/table.

9.3 Referencing and Citations

Whatever your project, there will be efforts of other people that will be a basis for your work. Most likely, there will also be efforts with an agenda similar to your own. As a scientist, you have to:

- acknowledge these contributions, and
- put the specifics of your approach into the scientific context.

This is achieved by proper referencing of all the things upon which your project relies and also all things that have inspired it.

A document with proper referencing allows the reader to: verify claims made in the text, find further background information when needed, and understand how the project fits in with contemporary developments.

Two things are needed: citations in the text and a reference list. Citations in the text shall adhere to some *Citation Style*. Which citation style you choose is of little importance, but sticking to the chosen one is important. Common choices within the department are IEEE and Harvard.

The reference list allows the reader to find the supporting material, but also to make a quick preliminary evaluation of the quality of the material.

There are tools that will help you manage references and citations. Reference managers, for example, Zotero, reduce workload by storing the information and providing typeset citations and references when needed.

9.3.1 Writing references

Supporting material comes in many different formats: books, academic journal papers, newspapers, lectures, web sites, etc. Referencing them requires different sets of bibliographical data. The base information needed is the *author*, by whom it is *published*, and *when* (usually, just year).

The bibliographical information expected for selected kinds of sources is tabulated in Table 2. The table is based on the popular BibTeX format for storing bibliographical information.

Why all this information, when we can just give a URL?

- It gives the reader information which the URL does not give. The reader may recognise the name of the author, or that the journal is reliable.
- URLs change or disappear.

By all means, provide the URL (or even better the DOI) if you have it, but do not omit any of the traditional information.

There is a huge difference in how authoritative a source is. It is expected that the majority of your references are from books, academic journals, and other academic writing, as these are peer-reviewed. This does depend on the topic of the project, so exceptions from this are possible.

Table 2: Selection of standard BibTeX entries and field types.
Key: + Required fields, o Optional fields, * At least one of

	article	book	inbook	incollection	inproceedings	tech report	unpublished
address		o	o	o	o	o	
author	+	*	*	+	+	+	+
booktitle				+	+		
chapter			*	o			
edition		o	o	o			
editor		*	*	o	o		
institution						+	
journal name	+						
note	o	o	o	o	o	o	+
number	o	o	o	o	o	o	
pages	o		*	o	o		
publisher		+	+	+	o		
title	+	+	+	+	+	+	+
volume	o	o	o	o	o		
year	+	+	+	+	+	+	o

For a book, such as the "LaTeX" book by Lamport we can see from Table 2 that we require the *author* (or *editor*), *title*, *publisher* and *year*. Thus, a BibTeX entry for this book may look like this:

```
@Book{lamport94,
  author =       {Lamport, Leslie},
  title =        {LaTex},
  publisher =    {Addison-Wesley},
  year =         {1994},
}
```

In a printed version this may be typeset as:

[Lam94] L Lamport: *LaTeX*. Addison-Wesley, 1994.

If you are not using BibTeX (or similar) you can just enter this final form directly (though you may well end up entering it more than once in the different documents you write).

9.3.2 References to Web-only sources

Be careful with references to the Web.

1. Web pages are not peer-reviewed and therefore lack the quality check that academic writing usually enjoys. Thus, errors and even complete fabrications are more likely. Due diligence is necessary.
2. Web pages change or even disappear. There is no guarantee that a reader will find the same information as the writer.

When referencing web pages always include the author, the title, and the date when they were last accessed.

9.4 Structure of Document

The basic structure of documents is:

- title page
- table of contents
- introduction
- several sections of main content

- summary or conclusion
- reference list
- possibly appendices

Please use provided templates.

10 Giving Presentations

Students will be giving presentations at Gregynog, but this is a transferrable skill that will be useful in many professional settings as well.

We will assume that the presentation will be given with the support of slides shown on a large screen. This has become the standard in many settings and is probably the easiest way to do it. We cannot guarantee that other tools, such as whiteboards, flipcharts or interactive screens are available, so we recommend the use of slides.

10.1 Talk Supported by Slides

The presentation is not the content on the slides. The slides are there to support the **narrative** told by the speaker. The main task of the slides is to

- emphasise important points,
- supply information that is visual (diagrams and images),
- help the audience follow the structure of the talk, and
- provide prompts to the speaker.

10.2 Slide Design

The important thing is obviously that the content selection is good and that the slides support the talk. However, we list some comments on design that you may want to keep in mind.

Font Use a sufficiently large font. Is it readable from the back of the room?

Text colour Ensure that the contrast to the background is sufficient. Remember that the colour you see on your screen may differ when projected by certain projectors, good contrast is needed.

One topic per slide Do not mix unrelated content.

Amount of information Do not put too much information on the same slide.

7 × 7 rule Do not have more than 7 lines on a slide. Do not have more than 7 words on a line.

Keywords / Key phrases Unlike normal writing it is often better to use single words or sentence fragments.

Diagrams and images Are *very* good if they carry relevant information.

Content first Decorative images and backgrounds may look pretty, but overused, they can distract from the main message.

10.3 Organising the Talk

Start with a title slide giving the name of the speaker and the title of the talk/project.

As the talk is rather short, an overly rigid structure with table of contents, section dividing slides and so on, may seem excessive. However, it is long enough that some clues are needed to help the audience follow the structure of the talk, so some selection of the above may be appropriate.

Include a summary at the end to reiterate the things that you want the audience to remember from your talk.

10.3.1 What to present

Before selecting the material to present you should always check what content is expected. This means that you should check what the brief for the talk is.

Looking at the brief in Section 7.2, we can see that we need motivation, aims/objectives, technical information on tools and techniques, and progress. Make sure not to miss things. But otherwise, it is just about finding a good balance.

It is commonly the case that it is difficult to fit everything in. Thus, we will often have to prioritise. Perhaps we find that we don't have time to present both aims and objectives in detail. In such a case we need to make a choice based on what is most useful to the audience. Another shortcut is that by showing off good knowledge of the tools and by having a good plan for attacking the problem, then we have done most of what is needed to show good progress.

10.3.2 The time issue

Presenting a content slide takes anywhere from 30 seconds to 3 minutes. (You did not put too much information on them, did you?)

10.4 The Performance

Think of the presentation as a stage performance. What can I do to make the presentation as engaging as possible? Consider: voice volume, eye contact, body language, etc. Include sufficient variation to avoid boredom. The purpose is not to be an entertainment act, but to attract the attention of the audience. Communicating with an attentive audience is much easier, and communicating information about your project is really the target of the presentation.

10.4.1 Preparation

It is not easy to predict how long something will take to present. Therefore you should practise your presentation and time yourself.

It is common that even a well-rehearsed presentation deviates from the time plan during the actual presentation. As mitigation it is good to prepare places in your talk where you can insert or remove material during the talk.

Practising your talk will also help you be freer in your actual presentation because you will know your talk better. It is useful to practise with an audience of volunteers where you can get some feedback, but there is also value in doing it by yourself.

Try to think about potential questions that can arise. This has two benefits: it will prepare you for the questions coming, and it may also highlight gaps in your presentation that can be filled before the presentation. Common questions include "Why did you choose this method?" and "How will you evaluate success in your project?"

10.4.2 The event

The environment you present in will be friendly. We know this from experience. We can guarantee it! Students will enjoy listening to your talk and, hopefully, you can feel like you are just presenting your project ideas and content to a friendly group of people – which you will be.

Everyone is nervous to some extent. This is to be expected and nothing to be alarmed about. Rehearsing, especially the start of the talk, may help. The good news is: it is less visible to the audience than the speaker thinks.

Be polite: greet people, answer questions kindly, and thank the audience for their attention at the end.

As an audience member you should also be polite. Give the speaker the same attention that you would like to receive from the audience.

11 Giving Poster Presentations

Giving a poster presentation is quite different from giving a normal presentation. Talking directly to just one or two listeners allows for much more interaction. It is common that poster presentations are query driven rather than prepared talks. Although, it would be common to have a short introduction prepared.

The visitors at the stand will have different interests, and the speaker should be alert to the interests of the listener and make necessary adjustments. Do not take offence at not having time to present everything you prepared.

11.1 Selection of Material to Present

As with any presentation, check what the brief is asking for, see Section 7.5.

11.2 Poster Stand Layout

The stand has a board of about one metre squared where poster(s) can be pinned. It is possible to have one large or several smaller posters. Use a format that suits the presentation.

11.3 Poster Design

Much of the same design hints as for slides apply here, but with some adjustments.

Font Use a sufficiently large font. Is it readable from 2 m?

Text colour Ensure that the contrast to the background is sufficient.

Headings Meaningful headings will help the reader find relevant information.

Do not put too much information on the poster If it takes 5 minutes to read, then the audience is not listening properly to what is said during that time.

Keywords / Key phrases Unlike normal writing it is often better to use single words or sentence fragments.

Diagrams and images Are *very* good if they carry relevant information.

Content first Decorative images and backgrounds may look pretty, but overused, they can distract from the main message.

11.4 Demo

Most projects will have at least some part that is possible to demonstrate live. A running demo can significantly strengthen a poster presentation.

Prepare your demo carefully. Make sure that everything needed is present and that you are comfortable getting it to run on the day. Show off as many features as possible. Allow the audience to try things out for themselves.

11.5 Printing

At your own cost, you can print your poster(s) at professional print shops or use on-line services.

12 Some Hints

- Pick a project that you will find fascinating to complete.
- Develop a good working relationship with your supervisor.
- Start early and work regularly.
- Keep notes as you go.
- Always read the brief for every assessment multiple times.