AE2GRP Software Engineering Group Project

Student Handbook 2017–2018

(This version: September 27, 2017)

Dave Towey

Contents

Introduction 2
Important Dates 2
Groups 3
Supervisors 3
Project Descriptions 3
Class Sessions 4
Meetings 4
Equipment 4
Assessment 5
Assessment Guidelines 6
Peer Assessment 7
Revised Peer Assessment 8
Interim Group Report 8
Final Group Report 9
Software 10
Individual Report 10
Submission of Deliverables 11
Open Day 11
Presentation Day 12
Golden Rules 13

Introduction

During your first/second year of study you learnt some of the basic principles of computer programming. In industry, however, you won't usually be programming on your own, but will be working as part of a group of people developing a larger application. As an introduction to group working, part of your Part I Year is made up of a software engineering group project, in which you are divided up into small teams (groups) to design, program, and document a computer application.

The aim of the group project module is for you to gain some Software Engineering experience and experience in the various different aspects of group working. The former include engineering requirements, system architecture and design, user interface design, implementing a medium sized, multi-component system, systematic testing and debugging, programming in a team, and use of software engineering tools. The latter include running meetings, making collective decisions, time and people management, writing reports, giving presentations, interpersonal skills, and resolving conflicts. During the project, you will also learn that being a member of a programming team can be great fun!

The group project module runs for the whole year and is worth 20 credits.

Important Dates

The group project is assessed through a number of tasks to be completed at specific times during the project. There are also other important deadlines that must be met. The **TENTATIVE** key dates and deadlines for the academic year 2017–2018 are as follows:

Task	**Date or Deadline**
Equipment requests	From 12 October 2017
Group project site up and running	Thursday, 26 October 2017
Interim reports due	Thursday, 14 December 2017
Final reports and software due	Thursday, 19 April 2018
Open Day (TBC)	Wednesday, 2 May 2018
Presentation Day (TBC)	Wednesday, 2 May 2018

Groups

(The terms “group” and “team” may be used fairly interchangeably in this document.)

At the start of the group project module you will be divided into teams of around five or six students. The team division is done so as to ensure diversity and, to the extent possible, a well-balanced set of skills within the groups. Note that forming your own groups is not permitted, and that you are not allowed to swap places with someone from another group. Working with people from different backgrounds, most of whom you may not know initially, is a central aspect of this module, just as it is going to be in your future professional lives.

The tem allocations can be found in a separate document on Moodle. Note that it can take a couple of weeks from the start of the module before the groups are completely stable as it is not until then that the individual module choices are finalised. **Important!** If you are not on the list of students taking this module but should be, or if you have not been assigned to a team but know you should be in one, you need to get in touch with the module convenor as soon as possible. In particular, if you are an incoming exchange student, you are unlikely to be in a group from the start. Similarly, if you find that you cannot make contact with certain group members, perhaps because they have elected to do a different module, it is important that you let the convenor know.

Each team should appoint one member as the Team Leader. In addition to their normal work as a member of the team, the Team Leader will play a management role. For example, the Team Leader should ensure that everyone is involved, that work is divided fairly and according to individual strengths, that tasks are completed on time, and that nothing is forgotten in the project. This does not mean that the Team Leader decides everything. On the contrary, the project is “owned” by the team as a whole, and the team may also find it useful to (formally or informally) appoint other leaders for specific aspects of the project, like Chief System Architect, Repository Master, Lead Programmer, or Lead GUI Designer. However, the Team Leader has a specific responsibility to ensure that the group works effectively as a whole, that everyone pulls in the same direction, that the project progresses according to plan, and, ultimately, that the project goals are achieved and the prescribed deadlines met.

Choosing a leader can be difficult, particularly if the team members do not initially know each other. However, it is very important for the success of the project that a leader is chosen. Not doing this is one of the most common regrets found in the final reports from group projects of previous years. If you have problems deciding on a leader, your supervisor can help in making the choice.

Supervisors

Each team will be assigned a member of staff as their supervisor. The supervisor is **not** in charge of the project, and he or she is not there to make decisions for you. Rather, the role of the supervisor is to monitor the group, to ensure that it remains focused, to ensure that steady progress is being made, to promote discussion, and to offer advice on matters such as report writing and presentation. In some cases, the supervisor may offer technical advice. The supervisor also plays a significant role in marking the project. For some projects, the supervisor may also play the part of a customer for whom the application is being developed. Teams meet with their supervisor once a week: see meetings below.

Project Descriptions

Each supervisor will provide their group with a brief written description of the problem that is to be solved, typically a few paragraphs long. This document will form the initial specification for the application to be developed. Each group will have a different problem to work on. The initial problem specifications will likely be vague, so one of the first tasks of each team will be to understand their problem better, by discussing among themselves and with their supervisor, and then refine and expand the project description into a requirements specification that the team and the supervisor can agree on.

The problem specifications will be made available in Moodle.

Class Sessions

The module is supported by a series of class sessions given throughout both the autumn and spring semesters. The exact number and content may vary from year to year. Various practical aspects of the module are covered, there are usually specific support sessions on presentation and writing, and there may be guest lectures, e.g. from industry representatives, covering a wide range of topics relevant to software engineering, including professional and quality issues. See the Moodle page for details.

Meetings

Each team should arrange regular times for two kinds of weekly meetings: one formal meeting with their supervisor present, and one (or more) without.

The weekly meetings with the supervisor may be held in his or her office, and should last about 30 minutes. The purpose of these meetings is for the team to report on progress since the last meeting, discuss any problems that have arisen, and plan the work that will be carried out by each member before the next meeting. These meetings may be quite formal in nature, with a chairperson (who runs the meeting) and a secretary (who records the progress of the meeting and produces "minutes"). These tasks should rotate among the team members, so that everyone gains experience with both tasks. Most students will not have attended formal meetings before, so towards the start of the group project module you may be given a special lecture on this topic. This lecture gives some suggestions for how to structure your meetings and explains the roles of the chairperson and the secretary.

The weekly meetings without the supervisor will normally be held in a mutually convenient location in the school, or elsewhere on campus, and have no prescribed time limit. What goes on at these meetings is completely up to the team, but they are usually more informal in nature. There is no requirement for a chairperson or minutes, but some groups may find it useful to carry on with these practices here.

Equipment

In the first few weeks of the group project module, each group needs to think about whether they will need any **non-standard** hardware or software (list of standard software will be provided by the IS technical support group). If so, but only then, you need to make the equipment request, detailing the hardware and software requirements for your project, via the Computer Science technical support staff, Joseph Zhang (Joseph.Zhang@nottingham.edu.cn). Under no circumstances may students themselves install hardware or software on school machines. Note that you only can make **one** request related to your group project.

What kind of requests can you make? No guarantees are made that all requests can be accommodated, but Technical Support will try to help with any reasonable request. The most common would be installation of (various kinds of) “free” software that normally is not available on School machines, or standard hardware (like a PC) configured in some specific way for your project. Beyond that, it may be possible to provide access to hardware and software that the School happens to have anyway. Also, don't forget that as it was your supervisor who set the project, he or she might be the one with access to special hardware, or he or she might be willing to purchase something to support the project.

For further details, please discuss with your supervisor.

Assessment

The group project is assessed through a number of tasks to be completed at specific times during the project. There are no written examinations. Most tasks are carried out as a team, but there is one individual task: an Individual Report to be handed in at the end of the project. Consequently there are two parts to the final mark each student will get: one part for the team tasks and each individual's contribution to those, one part for the individual report. This is explained in more detail in the following.

For the group tasks, each **Team** is awarded a **Collective Team Mark** on the standard university scale, with the different aspects of the project contributing according to the following table:

Group Task	**Weight**
Team Project Site	2 %
Interim Group Report	18 %
Final Group Report	30 %
Software	20 %
Open Day	15 %
Presentation Day	15 %

Peer assessment is then used to derive an **Individual Mark for Group Work**, on the university scale, for each team member from the Collective Group Mark. This is done by **distributing** the collective mark among the team members **according to merit** as perceived by the group members themselves (but **vetted** by the supervisor), such that the average of the individual marks for the team work equals the assigned Collective Team Mark.

The Individual Report is also assessed on the standard university scale. The Individual Mark for Group Work and the mark for the Individual Report are then weighted together into an **Overall Individual Mark** according to the following table:

Component	**Weight**
Individual Mark for Group Work	80%
Individual Report	20%

Important! Late submissions, be it of printed copies or electronic copies of reports and software, will be penalised at the standard university guidelines of 5 % per working day.

Most tasks are marked by the supervisor and at least a second marker. The Interim Report is marked by the supervisor only. For the Open Day, all members of academic staff are invited to mark the displays in addition to the supervisor and second marker. For the Presentation Day, the presentations are usually marked by a third member of staff in addition to the supervisor and second marker.

Assessment Guidelines

The following guidelines are applied when judging the quality of the Group Reports and Software:

- *Exceptional (90–100 %)* The reports and software should exhibit all the characteristics of an Excellent grade. Additionally, the project should have been carried out in an utmost systematic and professional manner, as evidenced by a problem analysis and subsequent requirements specification of stunning clarity and insight, a system design of highest possible quality that manifestly meets all requirements and given at a level of precision and detail that directly could be translated into an implementation, an implementation of highest possible quality and completeness whose conformance to the original specification has been verified rigorously, and impeccable project management in terms of planning, workload management, meeting deadlines, making the best possible use of each team-member's skills and strengths, and with absolutely minimal (technical or otherwise) guidance from the supervisor.
- *Outstanding (80–89 %)* The reports and software should exhibit all the characteristics of an Excellent grade. Additionally, the project should have been carried out in a very systematic and professional manner, as evidenced by a problem analysis and subsequent requirements specification of significant clarity and insight, a system design of very high quality that manifestly meets all requirements and given at a level of precision and detail that more or less directly could be translated into an implementation, an implementation of highest possible quality and completeness whose conformance to the original specification has been verified thoroughly, and very skilled project management in terms of planning, workload management, meeting deadlines, making vary good use of each team-member's skills and strengths, and with very little (technical or otherwise) guidance from the supervisor. .
- *Excellent (70–79 %)* The reports and software should display a complete and thorough understanding of the conceptual and practical issues surrounding the project topic, including related work. The project should have been carried out in a systematic and professional manner, as evidenced by a clear and insightful problem analysis and subsequent requirements specification, a system design of high quality that meets all requirements and substantially provides enough detail for implementation. Software should be completed in all respects and exhibit very high quality. There should be evidence of a high degree of testing. Supporting documentation should be complete and approaching the standard of high quality professional documentation.
- *Good (60–69 %)* The reports and software should show a good understanding of the conceptual and practical issues surrounding the project topic, including an adequate grasp of related work. The quality of the analysis, requirements specification, and design should be good, and the writing of the reports should be good in general. Software should be competently written. Evidence of testing should be presented. The software should be a complete and usable package which not only illustrates the principles of the work but also exhibits good levels of quality. Supporting documentation should be excellent for all purposes; it should be complete, well written, well presented and generally exhibit high quality.
- *Average (50–59 %)* The reports and software would be expected to display an adequate understanding of the key conceptual and practical issues, although weakness may be present in some areas. Some account taken of related work. The quality of the analysis, requirements specification, and design would exhibit significant weaknesses; the design would not as it stands constitute an adequate basis for implementation. The writing would exhibit some flaws. Software should be adequate to illustrate principles; it may display weakness in areas not central to the work and lack comprehensive testing. Supporting documentation would be well presented yet lack completeness; the quality of the documentation should be very good.
- *Adequate (40–49 %)* The reports and software would display an incomplete understanding of the central issues relating to the project topic. The reports would lack a clear structure and strong argument, and the quality of analysis, requirements specification, and design would be below average. The writing would be mediocre. Software would be incomplete, poorly commented and difficult to understand; it would exhibit poor levels of quality. Supporting documentation would be adequate.
- *Poor (below 40 %)* The reports and would display a very poor understanding of the project area; there would be no clear structure and the analysis may be weak or incomplete. The reports would be poorly written and presented. Software would be limited in capability, and difficult to use. Supporting documentation would be inadequate for most purposes.

The Open Day is marked primarily on

- how well demonstrations work
- the overall quality of the display
- the professionalism of those manning the stall (attitude, ability to demonstrate and explain)

The Presentation Day is marked primarily on

- the quality of the presentation
- how the team handles questions and answers

The Individual Report is primarily marked on

- the quality of the reflective statement
- the overall qualities as a report such as:
  - writing
  - structure
  - typesetting

The Individual Report is *not* marked on the contribution of an individual: peer assessment takes care of that. But showing a clear understanding of ones role in the project, the true value of ones contribution, and what was learned about working as a team, including an honest self-assessment, is important.

Peer Assessment

As explained above, **peer assessment** is a key aspect of the overall assessment. It is used to distribute the collective group mark to each individual according to merit as perceived by the peers, i.e. the other team members. Each team member is asked to rate each of his or her peers according to a number of aspects on a purpose-designed form. A written justification supporting the ratings is also required for each member. The peer assessments are submitted in **complete confidence** as part of the Individual Report, and will not be seen by the other students. Note that, **under certain circumstances**, it is possible to **revise** the assessment up until the end of the Presentation Day. See below.

The COMP2043.GRP Peer Assessment Form will be available on Moodle

Be sure to read the instructions on the forms carefully, as well as the instructions below, before filling out the forms. The completed forms are to be included as an **appendix in your Individual Report**.

Each group member is assessed according to the following aspects:

- Research and information gathering
- Creative input
- Co-operation within team
- Communication within team
- Quality and quantity of concrete contributions to team deliverables
- Attendance at meetings

Thus it is not just the amount of contributed work that is being evaluated, say in terms of words in the reports or lines of source code, but a wide spectrum of contributions all of which are important for a successful end result. This also means that everyone gets a chance to contribute according to their specific skills and get proper recognition for this. Clearly, someone who excels in designing the system architecture and implementing it has made a very important contribution. Likewise, someone who is a great writer and took the lead on getting the team reports together has also made a big contribution. But someone who was instrumental in mediating between conflicting views, say, and thus helped holding the team together has clearly also made a very valuable contribution to the team as a whole.

Each aspect is rated on a five-point scale ranging from "None" via "Adequate" to "Excellent", where **"Adequate"** means that the assessed person has done what is **expected** for that aspect: no less, no more. Note that it is the **relative** performance of group members that matters as the peer marking only serves to redistribute the assigned collective group mark. For example, if everyone rate everyone else "Excellent", the end result is that everyone gets the same mark for their contribution to the group work which is going to be equal to the collective team mark.

Assessing your peers is a privilege and big responsibility. Be fair and objective in your evaluations. In the past, the members of the School Faculty have been very impressed with the quality and honesty of the peer assessment, and there have almost never been any problems. However, each supervisor is charged with carrying out a sanity check on the peer marks based on what they have learnt during the interaction with the team members throughout the year, as well as from the team and individual reports. Should there be obvious problems with some of the peer marks, then the supervisor together with the module convenor will override any or all of those marks where necessary.

Revised Peer Assessment

Note that the initial peer assessment is submitted as part of the individual reports. This is usually sometime before the Open and Presentation days, so the majority of work will have been completed by then, and the time of report submission is thus a reasonable time to carry out the peer assessment.

However, it is conceivable that the performance of a peer during the final preparations for and at the Open and Presentation days is **significantly** better or worse than his or her previous performance, so much so that the initial assessment becomes very unfair. Here "very unfair" means that the **overall change of the assessment must be at least 3 points** (three different aspects changed one step up, one aspect changed two steps down and one other aspect changed one step down, etc.). Should this be the case, it is permissible to revise the assessment for a peer **one time**, up until the end of the Presentation Day. The revised assessment is subject to vetting by the supervisor and module convener just like the initial one.

To revise an assessment, the following procedure must be followed:

- The original assessment form for the peer in question should be amended to show the new assessment. It should also state that this is a revised assessment. Further, **each** individual change must be justified in writing by **adding** a paragraph that states the old assessment, the new assessment, and why it has changed.
- The revised form must be e-mailed in **PDF format** to **both** the supervisor of the group and to the module convener no later than 18:00 on the Presentation Day.

Interim Group Report

The Interim Group Report should be 4000–5000 words (around 12–15 pages; excluding any appendices). It is due around the half-way stage of the project. Each team must submit one such report, written as a group. The main content of the report should be:

- Updated and expanded description of the problem to be solved.
- Background information and research such as
  - survey of any existing systems that address similar problems
  - results of any market research conducted
  - results of technical research into suitable platforms, tools, technologies, algorithms, data structures, etc.
- Requirements specification for the system to be built (agreed between the team and supervisor).
- Initial design of the proposed system and its user interface.
- Record of key implementation decisions, such as programming languages, operating systems, computers, and any additional software and hardware to be used, along with reasons for those decisions.
- Results of any initial implementation steps/prototyping.
- Discussion of any problems encountered so far, including both technical issues and management issues, like group working etc.
- Time plan for the project.

One printed copy of the interim group report should be submitted as well as an electronic copy in PDF format. The front page of the report should include:

- The text “COMP2043.GRP Interim Group Report”
- Project title
- Date
- Team name/identifier
- Names and School of CS usernames of all group members
- Name of supervisor

Final Group Report

The Final Group Report should be 7000–8000 words (around 20–25 pages; excluding any appendices). It is due at the end of the project. Each team must submit one such report, written as a group. The report should be a self-contained, updated, and expanded version of the Interim Group Report less parts that are no longer relevant.

Important! The report must be self-contained. In particular, the report must not assume that the reader has read the Interim Group Report prior to reading the final report.

In particular, the Final Report should include:

- Updated design of the system and its user interface.
- Discussion on the implementation and testing of the system. This must include a list of all major system components, which of these were written by the group and where the others come from, and an overview of the developed source code hierarchy.
- Summary of what was achieved, referring to the stated requirements.
- Reflective comments on the success of the project, both from a technical and a project management perspective, including group working issues etc.
- An appendix giving a description of how the developed system was tested (test cases, example outcomes, etc.)
- Minutes: Minutes from all formal meetings should be submitted as an appendix.
- A user-manual (if appropriate) should be included as an appendix. Excerpts of the developed code can be included in the report for illustrative purposes, but any lengthy excerpts should go into the appendices.

Two printed copies of the final group report should be submitted as well as an electronic copy in PDF format. The front page of the report should include:

- The text “COMP2043.GRP Final Group Report”
- Project title
- Date
- Team name/identifier
- Names and School of CS usernames of all group members
- Name of supervisor

Software

An electronic copy of all developed source code should be submitted at the same time as the group report at the end of the project. The code should be submitted either in the form of a zip archive or a gzipped tar archive of the entire source code hierarchy. Auxiliary components, such as code libraries developed by others, databases, or other lengthy pieces of data do not need to be included. Consequently there is no requirement what is submitted has to be complete enough to run “out of the box”. But, if feasible, it is good if what is submitted is complete and if instructions for how to run the system are included.

Important! All source code should be properly attributed. Code written by the team should be clearly marked as such (e.g. by a comment at the top of each file). If code has been adapted by the team for the project then the sources must be clearly identified. Any system components not written by the team (e.g. libraries beyond standard libraries) must be properly identified.

Assessed aspects of the submitted software include, but are not limited to:

- Functionality and features
- How well the developed software actually works
- The size and difficulty level of the addressed problem(s)
- The technical sophistication of the developed software
- How well the software is written (architecture, proper modularisation, proper abstractions, naming conventions, layout, documentation, etc.)
- Quality of the testing

Evidence taken into account include the submitted software itself, information from appropriate parts of the submitted reports (including the testing appendix of the final report), Open Day demonstrations, testing carried out by the first and second markers themselves during e.g. the Open Day, insight gained from the team presentations and ensuing discussions during the Presentation Day.

Individual Report

Individual reflection on the project is important for getting the most out of it. This is the primary role of the Individual Report. Additionally, this report contains each student's assessment of his or her peers. Each student must submit one Individual Report at the end of the project, written by his/herself. It should be 2000–2500 words (around 6–8 pages), excluding the peer-marking part and any appendices.

The main parts of the Individual Report are:

- Summary of own individual contribution to the project.
- Reflection on the project, the running of it, and own role within it, including honest and insightful self-assessment.
- Peer assessment.

The reflection should cover aspects such as what was achieved, what was not achieved and why, critical discussion on the running of the project, what you have learnt from the project, and what you would do differently if you were starting over again.

The peer assessment is carried out on a dedicated form. The form, in various formats, is linked from this document. Each student must evaluate **all members** of the group **except themselves**. Additionally, each evaluation must be justified by a written statement on the performance of each peer. See Peer Assessment for further details, including links to the evaluation form in various formats to facilitate the inclusion of the completed forms with the Individual Report. (If none of the supplied formats is suitable, printed, completed forms can be scanned for inclusion.)

Note that Individual Report is marked primarily on the quality of the reflection and on the qualities as a report, not on the extent of contribution as such to the project: see Assessment Guidelines above.

Two printed copies of the Individual Report should be submitted as well as an electronic copy in PDF format.

The front page of the report should include:

- The text "COMP2043.GRP Individual Report"
- Project title
- Date
- Team name/identifier
- Your own name and School of CS username
- A list, clearly labelled "other group members" of the other members of the group
- Name of supervisor

Submission of Deliverables

Please check the deadlines for each deliverable carefully.

Recall that the individual report must include the peer assessment as an appendix. Thus, peer assessment should **absolutely not** be submitted in separate files.

Each deliverable should be handed in **no later than 4:00 pm** on the due date. Printed copies should be handed in to the Faculty of Science & Engineering Office. Electronic copies are submitted through Moodle.

You are also strongly advised to avoid handing in at the very last minute, as any delays due to e.g. long queues to the printers or heavy load on the servers are entirely your own responsibility as such problems are not unlikely and thus should be taken into account when planning.

Open Day

The group project Open Day is a chance for the groups to show off their finished applications to the other groups, to students from other years, and to members of staff. We take over the labs for a day, and each team is given a small area in which to set up a "trade stall" with posters, leaflets, and a live demonstration of their application. Note that it is the overall quality and professionalism of the stalls that is assessed during the Open Day, not just the application itself. To make the day more fun, everyone dresses formally for the Open Day.

Before the Open Day each group should complete an equipment form detailing the hardware and software requirements for the Open Day. The form will list the standard facilities that are available for the Open Day, as well as the extra hardware and software that can be installed by the support staff on specific machines for use with specific projects. Under no circumstances must students themselves install hardware or software on school machines. To confirm that the requests are appropriate, this form must be signed by the supervisor. No guarantees are made that all requests can be accommodated, but the support staff will do their best to ensure that equipment is allocated fairly.

To give you a flavour, here are some photographs from the Group Project Open Day on 6 May 2015

Presentation Day

Each team must prepare one presentation, which is usually performed by the best speaker in the group. The presentation should be 10 minutes long, with an additional 5 minutes for questions. It is attended by both staff and students. The main content of the presentation should be a description of the problem addressed by the project, an overview of the application developed, and some reflective remarks on the success of the project. The presentation is usually given using electronic slides (e.g. Microsoft PowerPoint). Some groups may choose to include a live demonstration of their application, but, with the possibility of unforeseen technical problems, this is risky. We generally recommend that demonstrations are given using pre-prepared slides. Note that it is the quality of the presentation itself that is assessed, not the quality of the project or the application. To make the day more fun, everyone dresses formally for the group project presentations.

Many students may not have given presentations before, so prior to this part of the group project there is a support session on this topic. This session gives some suggestions for what to put into your presentation, what to put onto your slides, and how to speak well.

Before the presentations, each team should complete an equipment form detailing the hardware and software requirements for the presentation if there are any non-standard needs. The form will list the standard facilities that are available for presentations, as well as the extra hardware and software that can be installed by the support staff if requested. Under no circumstances must students themselves install hardware or software for presentations, or modify the settings of the presentation equipment. This form must be signed by the supervisor. Teams using anything other than overhead projector or data projector will be required to demonstrate in advance to the support staff that their presentation works properly in the presentation room.

The presentation day is divided into sessions, usually with three presentations in each. You will not be required to attend for the entire day, only for the session to which you have been allocated. The project presentation schedule will be available two to three weeks before the Presentation Day. You should read carefully the information given and check to see when and where your group will be presenting. The sessions normally start on the hour, and teams are required to arrive during the preceding break to make all necessary preparations (such as copying PowerPoint slides to the main computer and ensure they display properly) before the session starts to ensure a smooth and speedy transition between the presentations during the session. Here is a photo of an AE2GRP presentation session on 6 May 2015.

Golden Rules

Below are some tips based upon the reflective comments in the final reports from previous years group projects. Read them carefully; they are the golden rules for a successful project.

- *Work consistently throughout the project.* Start work on your project quickly, and don't leave all the real work until just before the deadlines. The group project module is worth 20 credits, each of which corresponds to 10 hours of time according to the official University Guidelines, so you should expect to spend around 200 hours in total on your project, which averages out to around **9 hours per week** during term time. If you are not spending this amount of time per week on your project, then you are not working hard enough, and will have problems later on.
- *Make sure that you have enough to do.* Don't wait to be assigned tasks - volunteer for things that you find interesting. Keep in mind that your peers' assessment of you will have a major impact on your final mark, so make sure that you are seen as a valuable group member by taking an active role in the project, striving to be co-operative and keeping the interest of the group and the project foremost, and working hard.
- *Don't leave the programming until too late.* It is very easy to spend too long designing your application, and then find that you don't have enough time to complete the programming. In particular, there are often problems linking the different parts of applications together. Make sure that the interaction between parts is well documented before programming begins in earnest, and don't leave linking the parts together until the last minute. Aim to have a prototype up and running as soon as possible, and certainly by late November. If you do not have *any* code running by then, you should probably be worried. Prototyping is also a great way to learn about what you are doing, and can thus have a significant impact on the design of the project. Don't be afraid to throw away a prototype once it has served its purpose and you understand how you should have approached the problem.
- *Do use proper tools to avoid wasting time and effort.* In particular, do use a version control system.
- *Don't think that you can do the write up in just a few days.* On average, 60 % of the mark is for the written reports.
- *Keep track of everyone's progress.* Make sure that everyone in your group is working consistently on the tasks they have been assigned. Your supervisor can provide some guidance with managing your project. Ultimately, however, it is the responsibility of the group themselves to monitor the progress of each member, to ensure that tasks are being carried out as required, and to take remedial action as and when appropriate.
- *Take the meetings seriously.* Without proper formal and informal meetings each week, your project has little chance of being successful. Spend some time preparing for each meeting, and make your own notes during the meeting in addition to the minutes. Note, however, that most of the real work on the project will be done outside of the meetings. Students whose only work on the project is to contribute to the meetings will find that they have little to write about in their reports, resulting in poor marks for the module.
- *Keep good records of your meetings.* Collecting your minutes together on a web page is a good way to ensure that they do not get lost, and that everyone can read them. Use the minutes to ensure that the group doesn't discuss the same points for weeks on end, that decisions made are not forgotten about, and that real progress is being made.