Student Handbook
SFT3500/SYS3030:
Industrial Experience Project

Department of Software Development
Monash University

Version 1.01

Tim Menzies
timm@insect.sd.monash.edu.au

Sylvia Tucker
stucker@fcit-c1.fcit.monash.edu.au

February 29, 1996

Contents

List of Figures

1 Introduction

1.1 Welcome

In this subject, you will build real-world software for real world clients. For most of you, this will be a somewhat of a challenge. The project work of this subject is far less structured and far less predictable than the other subjects you are used to. You may find that you are working long hours[1] on novel tasks. But take heart! When students look back on their time here at Monash, they often remark that the most practical and worthwhile thing they ever did in their undergraduate work was this industrial experience subject. This subject, they frequently observe, was the one that best prepared them for their career in the information technology industry.

The rest of this handbook describes the structure of the subject, its deadlines, and all its associated paperwork. Before we go all formal and official, we just want to say that we hope that this subject will be an exhilarating experience for you. When we look back at previous years, we see that the overwhelming majority of the industrial experience projects have been great successes. So, even though we will wish you good luck, we know that you won't need mere luck to complete SFT3500/SYS3030. Perspiration, maybe. But luck, definitely not.

By the way, if you think that there is too much paperwork and reports associated with SFT3500/SYS3030, then consider:

- What we ask you to do here is *much less* than you would be required to do in most real-world projects.
- All these reports are our safety net for catching out-of-control projects. We want you to have the satisfaction of building useful software for real clients. We need to help you avoid certain pitfalls of software development. We therefore ask you to fill in these reports since that will let us carefully monitor your progress.

Good luck!!

Tim Menzies
Sylvia Tucker

[1] In Section 7.2.3 we guess-timate that each student will spend, on average, up to 11 hours per week on this subject from now right through to the end of semester 2.

1.2 Acknowledgements

This document was originally based on material developed at Monash University by Rob Hagan (1985), Henry Linger (1989), David Foott (1994), and Brett Hodgson, Ann Gilbert & Helen Smith (1995); and at Newcastle University by Bob Cohen (1995).

1.3 Change History

Version	When	Who	Comments
1	27/2/96	Tim Menzies Sylvia Tucker	
1.01	29/2/96	Tim Menzies	Section 3.2; Interviews changed to 10/20 marks in S1/S2. Better time graphs example (Section 7.3).

2 About This Subject

2.1 Aim

To give final year computing students the opportunity to apply the skills which they have learned so far in the course to the development of real world systems.

2.2 Objectives

On completion of this subject students will have knowledge of :

- all stages in the process of developing a software system,
- the roles and responsibilities of clients, system users and developers in a system development project;

have an understanding of:

- the way in which software systems are developed,
- the levels of abstraction at which a system can be described,
- the role of tools, techniques and methods in the development of a software system;

have the skills to:

- plan and manage the full range of activities in a software development project,
- work productively in a team,
- communicate effectively with clients and system users;

and have developed attitudes which allow them to:

- work productively in a team,
- recognise the applicability and limitation of using particular approaches to certain problem areas,
- appreciate their social, ethical and professional obligations in designing, maintaining or advising on software systems.

In meeting the above objectives, emphasis will be placed on the quality of the product. This entails the use of proper techniques and tools that enhance product quality including appropriate analysis and design tools, quality reviews, appropriate software testing and a high level of user involvement.

The quality of the product is not judged solely on the merits of the software but also is dependent on packaging. This means producing all necessary documentation. The documentation must be well written and professionally presented. Further, it must be produced at the appropriate point in the development of the product.

The development of products will be oriented to computer solutions. However manual operations and/or organisational issues must be adequately addressed.

In addition to the products produced by the project the process by which these products are produced is an important aspect of the subject. These processes relate to how/when/why certain actions are undertaken by the group. It is important for the group to observe their process with the aim of improving it. For example, meeting process (group meetings, supervisor meetings and client meetings etc) - the products of these meetings relate to minutes, action lists and information for further work. The process relates to how you identified and recorded information, how you involved all people present, why you took a certain approach to an interview and whether it was successful. It relates to the group reviewing not just the products but also reviewing the process by which the product is produced.

2.3 Subject Requirements

2.3.1 Prerequisites

The prerequisites for enrollment in this subject is a pass in SFT2201 or equivalent.

2.3.2 Duration

The project runs from academic Week 1 to Week 26 inclusive of all non-teaching periods. Seminars will be held throughout the year during teaching weeks. It should be stressed that these are seminars and not lectures and hence groups and individuals are expected to actively participate by contributing experiences and asking questions.

3 Subject Structure

3.1 Assessment Grades

The mark for this subject will be “PGO” (pass - no higher grade available) or “NN” (fail).

Students will receive an “NN” for this subject if they fail to satisfy the following hurdle requirements:

- Attendance at all group, client, supervisor meetings.
- Attendance at the presentations and showcase.
- Organisation of a showcase.
- Project deliverables as agreed with the client.
- Failure to revise software as per reasonable client request (see Section 3.2).

See Section 6 for details on how to get a “PGO” mark.

The usual case is that all students in a group will get the same mark. However, it is possible for a group supervisor (see Section 5) to award marks to individual students.

3.2 Revisions to Software

Students are expected to deliver their software and final documentation (see Section 6.1.5) to their users by the end of October. After delivery, users may request changes. After consultation with the supervisor, some of those changes will be termed “reasonable changes” which students

Semester	Week	Area	See Section...
1	2	groups size and members finalised	
	3	groups must contact their supervisor by this week	
	4	legal documents signed and delivered	6.1.4
	5	slow-moving documents delivered	6.1.1
	11	initial set of fast-moving documents available	6.1.2
	11	presentation #1	6.4
	12,13	interview #1	6.5
2	3	technical review #1	6.2
	9	technical review #2	6.2
	12	presentation #2	6.4
	12	demo to supervisor	6.3
	13	interview #2	6.5
	1 week after exams	showcase	3.4

Figure 1: Schedule

must complete by the end of November. Note that the end of November is the absolute deadline for marking in order to give the office staff to prepare marks for the December 3 markers meeting.

3.3 Schedule

Students in the this subject must:

- Build a piece of executable software;
- Attend regular meetings with their clients and supervisor;
- Assess the software produced by other groups;
- And present their software:
  - Twice to the other groups;
  - Once in a public showcase.

Students should also attend a set of seminars on aspects of the software life cycle (see Section 3.5).

The subject schedule is shown in Figure 1. Note that most of the work must be completed before the semester two exam period starts.

3.4 Public Showcase

The public showcase is a big event. All the groups display their working systems in one large hall. The university encourages industry and members of the public to attend (the event is advertised in the major metropolitan newspapers). Second year students come along to see the sort of things they will be doing in third year.

Semester	Week	Topic	Joint?	Speakers Organised by. . .
1	1	Welcome and subject introduction	no	SFT
	2	Project & organisation management; acceptance documents	yes	SYS
	3	Conducting meetings, group dynamics (video)	yes	SYS
	4	Logical modeling and design	yes	COT/SYS
	5	None (Easter Friday)		
	6	OO methodology	no	SFT
	7	User-centred design	yes	SFT/SYS
	8	Physical design issues (GUI, prototyping)	yes	COT/SYS
	9	Quality assurance	yes	SFT
	10	Testing (test plan, acceptance criteria)	yes	COT/SFT
	11	Group presentations	no	n/a
	12	Security, backup, recovery and installation	yes	COT
	13	None		
2	1	Open panel: discussion on strengths/ weaknesses of various tools	yes	SFT/COT/SYS
	2	Acceptance and system testing	yes	COT
	3-10	External speakers, to be announced	yes	COT/SFT/SYS
	11	Preparation for showcase	yes	COT/SFT/SYS
	12	Group presentations	no	n/a
	13	None		
	Nov ?	Projects showcase	yes	COT/SYS/SFT

Figure 2: Seminar series

3.5 Seminars

Currently, the software life cycle seminars are planned for Fridays, 2-4pm. These seminars will be typically be joint presentations by lecturers from Software Development, Computer Technology, and Information Systems. See Figure 2 for the seminar topics.

3.6 Insurance

The University's Insurance does not cover personal accidents for students engaged in ”practical placement”. Project work undertaken in SFT3500/SYS3030 would fall within the term ”practical placement”. The Student Union does, however, carry such insurance cover, and students need not arrange separate cover for Industrial Experience. Information on the insurance policy is available from the union reception.

4 The Subject Co-ordinator

The subject co-ordinator has a background watching brief on all the groups. Problems which cannot be resolved at the supervisor-group level should be referred to the subject co-ordinator.

The 1996 subject co-ordinator is Sylvia Tucker.

5 The Supervisor

5.1 Role of the Supervisor

Each group will be assigned a supervisor for the whole year.

The role of the supervisor is *NOT* to act as analyst or designer of the project. That is the responsibility of the group. The role of the supervisor is to:

- Advise and assist the work flow of the group.
- Detect significant deviations from the project scope and time estimates.
- Provide an official high-level liaison point between the client and Monash University.
- Award marks to the group and to group members (see Section 5.2).
- Ensure that the groups are aware of how deviations from the reporting structure described in this document can affect their marks.
- If appropriate, the supervisor *may*:
  - Attempt to manage conflict resolution between group members and the client.
  - Make suggestions about down-sizing the initial project scope
  - Assist with complex negotiations with the client. However, for the most part, client liaison is the groups' responsibility.

5.2 Marks Awarded by the Supervisor

At the end of each semester, supervisors award a total of 30 marks to each group member based on an interview which explores their contribution to the project. The structure of these interviews is discussed in Section 6.5.

At the end of the year, the supervisor will award 15 marks as the *progress monitoring mark* if the paperwork associated with the project is produced appropriately; i.e.

- Each fortnight, a set of *progress summary documents* (defined in Section 6.1.3) must be prepared and placed in the pigeon holes of your supervisor and the IE project co-ordinator.
- The slow/fast moving design documents (see Sections 6.1.1 & 6.1.2) have been maintained diligently.
- The legal and final documents (see Sections 6.1.4 & 6.1.5) have been delivered at the appropriate times.

At the end of the year,supervisors award 30 marks at a live demonstration of the software (see Section 6.3).

Panels of supervisors will award 15 marks for the presentations in semester 1, week 11 and semester 2, week 12.

	Area	**Marks**	**Description**	See Section. . .
Group	Progress	15	A fortnightly report structure	6.1
	Peer	10	Technical reviews by other students checking for deviations of the product from the design	6.2
	Product	30	A demonstration of the working software	6.3
	Presentation	15	Two seminars on the project	6.4
Private	Personal	30	Each student's own contribution to the project	6.5
	Total	100		

Figure 3: Marking: the five "P"s

5.3 Agenda at Supervisor Meetings

Students will meet their supervisors on a schedule agreed on by themselves and the supervisor. Each such meeting will have at least the following agenda:

- Review the action list from last meeting.
- Record which students attended the meeting as well as any apologies for non-attendance;
- Discuss the progress of the project. This will include:
  - A general discussion of the current state of the project;
  - Reviewing any changes to the scope as agreed with the client;
  - Reviewing any changes to the slow and fast moving design documents (see Sections 6.1.1 & 6.1.2);
  - Reviewing the latest copy of the progress summary documents (see Section 6.1).
- Deciding a time for the next meeting;
- Prepare an action list of tasks that should be attempted for the next meeting. These tasks will be assigned to various people.

6 Assessment

Students will work in groups of five or six. A student will receive 70 marks (of 100) from an assessment of the group's work (see Figure 3).

The remaining 30 marks will be assessed from two interviews based on the structure of Figure 5. A "PGO" (pass) mark will be awarded if:

- Students satisfy all the hurdle requirements (see Section 3.1) **AND...**
- Students have an overall mark of 50 or more, **AND...**
- Students have earned at least 50% of the possible "private" marks; i.e. at least 15/30 **AND...**

- Students have earned at least 40% of the possible “group” marks; i.e. at least 28/70.

The private and group marks are shown in Figure 3.

Otherwise, a “NN” (fail) mark will be awarded.

On request, references for potential employers can also be generated from these marks.

6.1 The Progress Monitoring Mark

Project supervisors award the progress monitoring mark based on how well students have maintained the project-related documents. These project documents are divided into:

- The *slow-moving documents* (see Section 6.1.1) which will probably only need to be written once;
- The *fast-moving documents* (see Section 6.1.2) which will probably need to be revised regularly;
- The *progress summary documents* (see Section 6.1.3) which will be revised fortnightly.
- The *legal documents* (see Section 6.1.4).
- The *final documents* (see Section 6.1.5).
- And any other documentation as negotiated with by the user.

6.1.1 Slow-Moving Documents

When:	Version 1 is due by week 5, Semester 1
Copies to:	Supervisor, client
Revised:	As required. It is anticipated that the slow-movers will not be revised very often.

The slow-moving documents comprise:

- A short description of the **business case** for the system. The business case is like the executive overview of the system for the board of directors.
- Statement of technical requirements of the project; e.g. hardware, software, memory requirements.

It is anticipated that the slow-moving documents will be less than two pages long.

6.1.2 Fast-Moving Documents

When:	Version 1 is due by week 11, Semester 1. However, preliminary versions should be developed from week 4.
Copies to:	Supervisor and client
Revised:	As required. It is anticipated that the fast-movers will be revised often.

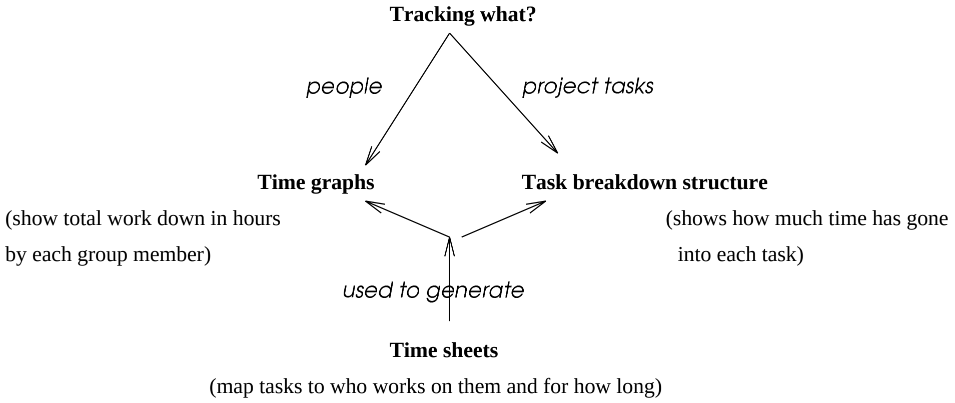
Tracking what?
people
project tasks
Time graphs
(show total work down in hours by each group member)
Task breakdown structure
(shows how much time has gone into each task)
used to generate
Time sheets
(map tasks to who works on them and for how long)

Figure 4: How to track a project

The fast-moving documents comprise:

- **Screen designs**, if applicable.
- **Design documents** in some agreed notation.
- **Loans list**: all materials on loan from the user.

6.1.3 Progress Summary Documents

When:	Version 1, Week 4, Semester 1
Copies to:	Subject co-ordinator and supervisor.
Revised:	Every fortnight.

The progress summary documents comprise:

- **Risk list** (see Section 7.1).
- **Task breakdown structure** (see Section 7.2) showing the tasks and their allocated time.
- **Time graph** (see Section 7.3) showing the cumulative hours spent by each team member. Note that all members of a group must sign the submitted time graphs to certify that they all agree that this graph represents how much work each group member has put into the project.

Maintaining these documents implies that students must keep individual time sheets. These timesheets map into the task breakdown structure and the time graph in the following manner:

- A project can be divided into *what* is done and *who* does it (see Figure 4).
- The task breakdown compares the time estimated and the time taken for the tasks of the project. The task breakdowns can be used to detect if the project is behind schedule.

- The time graphs shows how much time each student has put into the project. The time graphs can be used to detect if certain group members are doing more than their fair share of the work.

The progress summary documents will be used by the subject co-ordinator to compare the progress of the different groups. It is therefore **very important** for the groups to produce these documents in a standard format. Sample formats are shown in Section 7. Note that groups jeopardise their 15 marks for progress monitoring if they deviate significantly from the formats of Section 7.

6.1.4 Legal Documents

When:	Week 4, Semester 1
Copies to:	One original to subject co-ordinator. One copy to supervisor.
Revised:	Never
Signed by:	Students do not sign the agreement (see Section 8). After the client signs it, an original is passed back to the supervisor who organises signing by an authorised official of the University from the solicitors office. On the other hand, students are to sign and witness the deed (see Section 9).

The legal documents comprise:

- An agreement with two schedules.
- A deed.

Section 8 is the agreement which each team should complete and organise to be signed by the client in the presence of the project manager. The supervisor is to counter sign the document. The supervisor is responsible for explaining the need for such a form to the client and arrangement for the signatures. All signed forms should be passed on to the subject co-ordinator.

Where Monash University staff and students develop a system under the industrial experience program, full usage rights are granted to the host organisation. However, Monash University retains the copyright over the system. Where the host organisation intends to sell the system to a third party, a suitable agreement must be made between all parties.

It is the policy of the Faculty of Computing and Information Technology that staff and students should not be paid for industrial experience work. Students are therefore not considered employees of the host organisation. Students are reminded that on enrollment they agreed that all intellectual property belongs with Monash University. To ensure that students are fully aware of this, students must sign the **deed** in Section 9 acknowledging that they have no intellectual property rights over the software developed in this subject.

IMPORTANT NOTE: One deed must be signed by EACH student.

6.1.5 Final Documents

When:	Week 12, Semester 2
Copies to:	Client and supervisor
Revised:	Never

The final documents comprise:

- Listings of all source code.
- Source and object code in a machine readable form.
- Complete user documentation for all functions delivered.
- All letters, memos, agreements and any other formal correspondence.

6.2 The Peer Mark

When:	Weeks 3 and 9 of Semester 2

Twice in semester two, each group will anonymously review another group's work:

- In week 3 of semester 2, students will check that another group's project documents contain sufficient detail to describe the project. Peer mark #1 awards 3 marks.
- In week 9 of semester 2, students will look for deviations between the design documents and the current working system. Peer mark #2 awards 7 marks.

Groups performing the peer review will have one week to complete the review. The reviews will be anonymous; i.e. groups will not know who is reviewing their work.

6.3 The Product Mark

When:	Weeks 12 of Semester 2

Students will demonstrate the working system to their supervisors:

- Starting at 16 marks, 1 mark will be deducted for each major bug found and 0.5 marks will be deducted for each minor bug found.
- Supervisors will award another 14 marks for the quality of the system:

14	Exceptional
12	Outstanding
10	Very Good
8	Competent
6	Satisfactory
4	Improvement required
2	Unsatisfactory

  Note that the "quality" mark is subjective. If the supervisor feels that the software would not be useful to the client, then the quality mark may be very low.

6.4 The Presentation Mark

When:	Week 11, Semester 1 and Week 12, Semester 2

Students must present their project twice: semester 1, week 12 and semester 2, week 12. The first presentation gives a group a maximum of 5 marks while the second presentation gives a group a maximum of 10 marks. Presentation #1 presents the slow and fast moving documents. Presentation #2 summaries presentation #1 as well as showing:

- Screen dumps of the working system
- A live demo of the working system (if possible, and if you are feeling brave[2]).
- An evaluation of the final system. In particular, can it be shown that the project has not failed (i.e. has meet some minimal requirement(s)) and/or has actually succeeded (i.e. has excelled in some demonstrable manner).

Marks will be awarded by a panel of supervisors attending each presentation.

6.5 The Personal Mark

When:	Weeks 12 or 13 of Semester 1 and week 13 of Semester 2

Twice in the year, supervisors will conduct an interview with each student. The semester one interview will contribute 10 marks to the student's final mark. The semester two interview will contribute 20 marks to the student's final mark. The interview is structured according to Figure 5.

[2] Note a truism of software development: if you want to find the bugs in a program, demonstrate it live to the CEO.

Key Points	*S1*	*S2*	*Measures*	*Skill Emphasis*	*Possible Questions*
Subject Structure Undertake SE project and marking scheme	1	1	Understanding of subject schedule and marking scheme	Know project deadlines	“What is due, when and how many marks do you get for these?”
Project Role Undertake SE project and complete required tasks	2	4	Meets project objectives Demonstrates knowledge in key areas of project work	Achievement oriented Analytic thinking Initiative	“What are the deadlines in your project?” “What are your project deliverables?”
Communication Contribute at supervisor meetings Contribute/assist to reports, documentation	2	4	Peer impact Level of contribution	Communication Integrity	“Does the whole group know what is going on?” “Can each members of the group describe the project?” “Ok, Jamie, who is doing what in your group?” Point to any portion of the design document and ask: “What is going on here, then?” “Which documents do you write?” “What role do each of you have for the presentations?”
Customer Understand customer requirements	2	4	Customer knowledge	Adaptable Customer orientation Communication	“What does your client want and when?” “What is the basic business case?”
Project Management Contribute to project management. Awareness of key elements of project management.	2	4	Contribute to effective project management.	Planning Teamwork Analytical thinking	“What changes were made and why to the project schedule in response to the events that occoured during the project?”
Applied Technical Skills Develop expertise in SE skills - analysis, design, implementation	1	3	Demonstrated and applied technical knowledge and skills to the project tasks	Teamwork Initiative Achievement oriented Planning	“Describe the technical strengths and weaknesses of your software tools.” “Which parts were the most problematic?” “Would you recommend these tools for similar projects in the future?” “What tools would be better and why?”
Total	10	20			

Figure 5: The semester 1 interview gives 10 marks, the semester 2 interview gives 20 marks.

7 Example Documents

7.1 Risk List

7.1.1 About Risk Lists

Risk is everybody's business. Risks should *not* be hidden from the supervisor or the client. The risk list should be distributed to all team members and the client on regular intervals.

The risk list contains the top **ten** problems that the project is or may encounter in priority order (most worrying on top, least worrying on the bottom). A risk may be identified but may not currently be active. Such risks are called *passive* risks. Each risk is augmented by the action taken to minimise the projects exposure to this risk.

If you do not hunt for risks, then risks will come hunting you. The risk list serves to make team members aware that risks can and should be monitored and managed.

7.1.2 Skeleton Risk List

Priority	Active or Passive?	Notes
1-10	P or A	*Risk name* Action to minimise this risk

7.1.3 Example Risk List

Priority	Active or Passive?	Notes
1	A	*Team has never built a Visual Basic/Excel connection before.* A working prototype is to be developed ASAP.
2	A	*The west coast division may not supply the 1994 historical data by the time we need it.* Organise a tele-conference with the west coast manager immediately to ensure they are aware of our requirements.
3	A	*Team unfamiliar with the software being used.* Relatively small and relatively unimportant modules have been identified within the current design. Team members are developing their skills in the software by coding these small sections.
4	P	*Interface may not be what client wants.* Paper mock-ups of interface will be built. Typical business scenarios will be documented. In a session with client, the proposed interface will be exercised by walking through the scenarios.
5	P	*Other subject commitments may hinder development of system.* Team leader will collect information of exam and assignment deadlines from all students. Lots of work is planned for the non-teaching weeks.
...	...	...

7.2 Task Breakdown Structure

7.2.1 About the Task Breakdown Structure

The task breakdown structure divides the total project into track-able portions. Each line in a table can break down into a sub-table For example, in Figure 6, see how the *2. Design* line in the first table expands into a whole table in its own right (see Figure 7).

With each entry, groups must make an *initial* estimate for how long each task will take (measured in hours). This initial estimate is never changed. However, if the group re-estimates that task, then the new figure is entered into the *current* column. At the end of the project, students can get a feel for what aspects of their initial estimates were incorrect.

Based on **time sheets that every student will maintain**, the *To date* column can be maintained and reported as a percentage of the *current* estimate column. This percentage is shown in the *% completed* column.

A review of the *% completed* column can be used to quickly identify a project that is off course. In particular, when the *current* estimate is much larger than the *initial* estimate, and the *% completed* column is very low, then some task is clearly out of control.

Groups may be asking themselves, how can novice software engineers complete such tables? Well, the short answer is that students are not expected to be totally accurate in these estimates. However, only by committing to an initial estimate, then tracking how it changes, can student software engineers refine their estimation skills. Further, towards the middle of semester two, these tables will provide a very clear picture of who is nearly finished and who is in trouble.

Note that these task breakdown tables will most almost certainly change over the period of the project. New tasks will appear and the time required for existing tasks will change. The key thing is *not* to estimate correctly initially, but to react appropriately to the inevitable changes in the project structure.

7.2.2 A Skeleton Task Breakdown Structure

		Estimates		Actual	
ID	Task Name	Initial (hours) A	Current (hours) B	To date (hours) C	% completed D D=(C*100)/B
number	string	number	number	number	0-100
	Total				

7.2.3 An Example Task Breakdown Structure

Sample task breakdown structures are shown in Figures 6, 7, and 8. Note that Figures 7 & 8 are expansions of the activities associated with he *design* task of Figure 6. The task names and the times allocated in these figures may not be applicable to your project. For example:

- Figure 8 shows task 2.1.1 which divides task 2.1 shown in Figures 7 which, in turn divides up task 2 shown in Figures 6. Note that all tasks do not need to be divided up into three levels.

However, the numbers in these tables can demonstrate several things:

- Groups that record their time generally report that they spend between 1000 to 2000 hours on the project. Assuming 6 people in a group and 30 weeks working on the project, then each group member averages 11.1 hours per week on SFT3500/SYS3030. Note that unless this workload is spread out effectively throughout the year, then SFT3500/SYS3030 can take over a student's life in the second half of semester two. **You have been warned.**

- These sample task breakdown have several commendable features:

		Estimates		Actual	
ID	Task Name	Initial (hours) A	Current (hours) B	To date (hours) C	% completed D D=(C*100)/B
1	analysis	150	150	150	100
2	design	200	200	50	25
3	code	250	250	0	0
4	test	200	200	0	0
5	changes	180	180	0	0
6	user-training	100	100	0	0
7	progress documentation	60	60	5	0.1
8	showcase preparation	20	20	0	0
	Total	1160	1160	205	17

Figure 6: Top-level task breakdown structure

		Estimates		Actual	
ID	Task Name	Initial (hours) A	Current (hours) B	To date (hours) C	% completed D=(C*100)/B
2.1	excel/basic interface prototype	40	40	20	50
2.2	report subsystem	20	20	10	50
2.3	database design	40	40	10	25
2.4	screen design	30	30	10	33
2.5	checking screens with users	30	30	0	0
2.6	trialing screens on users	30	30	0	0
2.7	word-processing	10	10	10	50
	Total	200	200	50	25

Figure 7: Task breakdown structure for the design task of Figure 6.

- Testing is usually ignored in the task breakdowns of novices. However, note that a large portion of Figure 6 is devoted to *testing*. Further, this group has acknowledged in their plan that testing will inspire changes to the system. Hence the 180 hours allocated to *changes*.
- This group knows that there is lots of work involved in getting a good *progress monitoring* mark (see Section 6.1). Hence, they have allocated two hours a week to the task of updating the progress-monitoring documents (see Sections 6.1.2 & 6.1).
- This group has allocated nearly two days (20 hours) to the preparation of their public showcase presentation.

Note that task 2.1 *excel/basic interface prototype* is allocated 40 hours. This is a lot of time and should be further sub-divided into another table (e.g Figure 8). As a rough rule of thumb, divide-up all tasks that are longer than twenty hours[3].

[3] And, on the other side of the coin, don't bother recording tasks that are real short, say less than 4 hours

		Estimates		Actual	
ID	Task Name	Initial	Current	To date	% completed
		(hours)	(hours)	(hours)	
		A	B	C	D=(C*100)/B
2.1.1	develop evaluation criteria document	4	4	4	100
2.1.2	do the "advanced excel" tutorial	8	8	8	100
2.1.3	do it (horrible version)	12	12	12	100
2.1.4	do it again, better	12	12	0	0
2.1.5	apply evaluation criteria	4	4	0	0
	total	40	40	20	50

Figure 8: Task breakdown structure excel/basic interface prototype of Figure 7.

Sophisticated spreadsheet programmers could set up all the above as a set of linked tables with the group member's own timesheets down the bottom of the file. This approach would simplify the data entry and the maintenance of the fast-moving documents. However, given the small size of the groups, this may not be necessary. In fact, a simple pencil and paper system might be simpler and faster in the long run.

A standard mistake beginners make with task breakdown tables is that they write one big table like Figure 8 and never see the "big-picture" shown in Figure 6. We encourage groups to identify useful high-level abstract phases of their software life cycle.

7.3 Time Graph

7.3.1 About the Time Graph

The time graph (e.g. Figure 9) shows:

- the cumulative time for each group member during the 30 weeks of the project;
- the cumulative total time worked by the entire group;

The graph should be sized so that the top border is at the current estimate for the entire project. For example, Figure 6 tells us that a project will take 1160 hours. Therefore, Figure 9's y-axis goes up to 1160 hours.

7.3.2 Example Time Graph

Figure 9 shows a sample time graph. That graph shows that the team members are putting different amounts of work into the project:

- The *rock star*'s cumulative total is far greater than everyone else on the team. This member of the group is doing most of the work, perhaps because they have certain technical skills that are crucial to the project. Rock stars have to be managed very carefully and diplomatically:
  - The rock star may feel that they are carrying more than their fair share of the project.

  - Their extra experience with the system may mean that they make design decisions that the other team members do not understand.

  When faced with a rock star, group communication becomes very important:

  - The rock star must ensure that the rest of the team understands what they are doing.
  - The rest of the team has to monitor the rock star so that the rock star does not waste time on technical irrelevancies to the project.

- The *passenger* has worked far less hours than the rest of the group. We call them passengers since the rest of the team is carrying them along. There are at least two causes of "passengers"

  - Incorrect project structuring. This group would need to assess its work to see if the "passengers" are somehow being cut-out of the decision making and work practices of the group. In this case, the group should look to methods to include all group members.
  - Low motivation or low technical competency of the passengers. This is a problem that must be discussed with the supervisors.

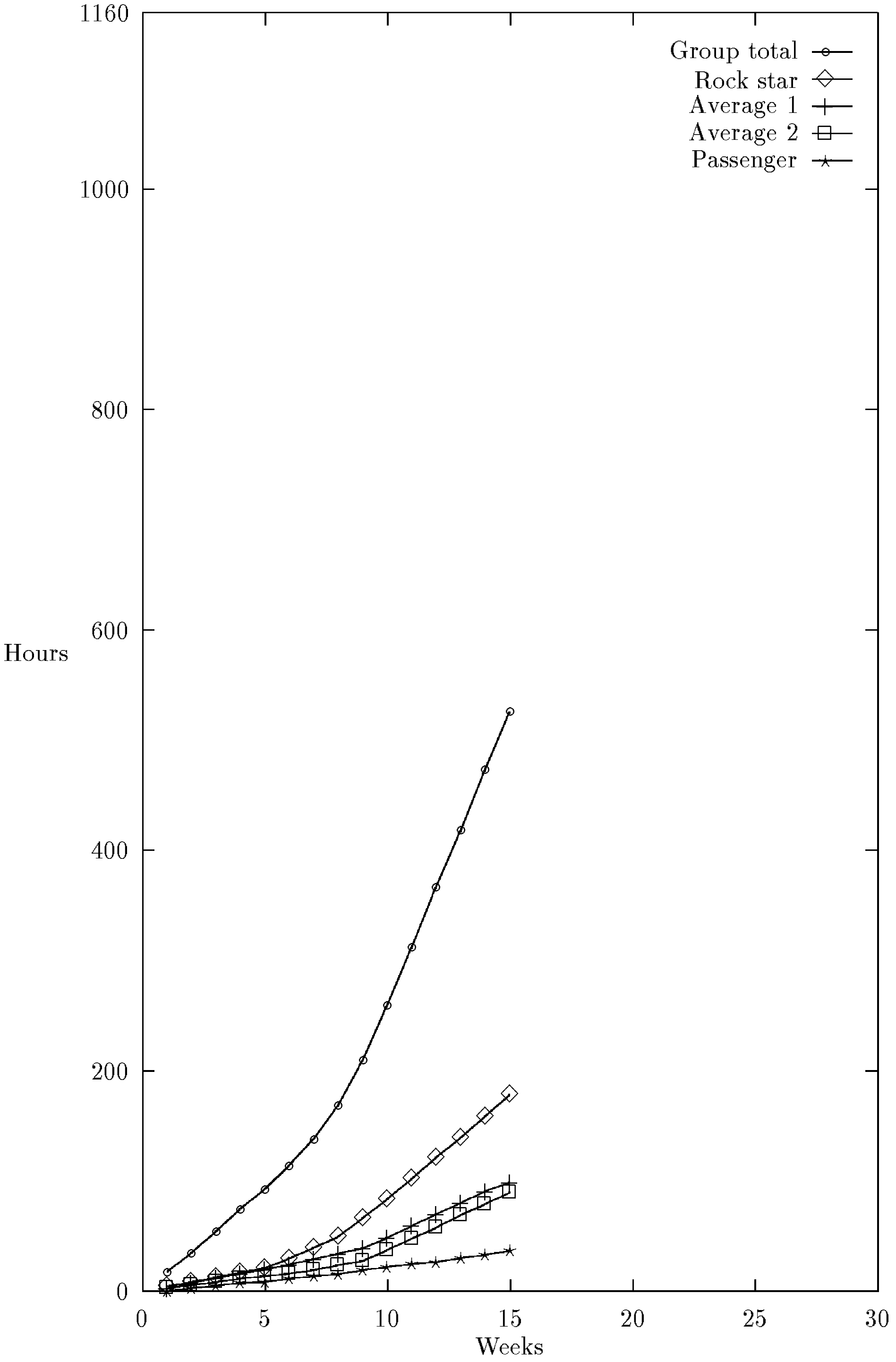

Figure 9: A sample time graph.

8 Legal Agreement

AGREEMENT dated this day of ,1996 between **MONASH UNIVERSITY**, of Wellington Road, Clayton, ("the University")

AND ____ ("the client")

of ____ .

1. A ________________________________ ("the system") will be developed during 1996 for the client by students named in Part two of Schedule 2 to this Agreement and as explained in Schedule l to this Agreement.
2. The students will be supervised by those persons named in Part 1 of Schedule 2 to this Agreement.
3. The system referred to in Clause l will form the basis of the students assessment in the subject SFT3500/SYS3030 (Software Development Industrial Experience Project) during l996.
4. The client acknowledges and agrees that all intellectual property in the system shall be the property of the University and that the client shall not be entitled to any property right or license in any copyright, patent, design or other intellectual property relating to the system other than as set out in this agreement.
5. The client warrants that it has the right and authority to grant the students and supervisors access to software used in developing the system.
6. The client agrees that it will not contest or challenge at any time hereafter in any legal proceedings or otherwise the ownership by the University of the intellectual property rights relating to the system.
7. The University will deliver to the Client a copy of the software and manuals created by the project team for the Client. The Client will pay the University a fair price for the storage medium in which the software and manuals are contained, or the Client will supply the University with substitute storage medium of the same type.
8. The Client is permitted by the University to use the system internally at no charge but not to sell or otherwise dispose of the system or any part thereof unless further agreed in writing by the University.
9. The University agrees that it will not independently market, sell or otherwise dispose of the system or any part thereof without the agreement of the client in writing.
10. The University does not warrant the system in any way and will not be liable for any loss or damage either direct or consequential suffered by the client arising out of the client's use of the system or any failure in the system.

11. The University shall not provide any maintenance or implementation support for the system beyond 1st December 1996.

12. Where the system involves enhancement to the client's existing software the client and the project team will together determine the scope of the enhancement. If the enhancement is deemed to be minor in relation to the overall software system, the University will waive all intellectual property rights to the enhancement. Otherwise all previous clauses will apply.

13. All information concerning the client's business including but without limiting the generality of the foregoing, client lists, software, circuit diagrams, hardware specifications, methods, systems, algorithms, and other elements created or developed by the project team on behalf of or learned from the client is confidential and proprietary information, and the project team will not duplicate, use or disclose any such confidential information to any person unless prior written permission has been given by the client.

14. The client shall not without the University's prior written consent use the University's name in any promotional material.

SIGNED by)
for and on behalf of)
("the Client"))

SIGNED for and on behalf of)
MONASH UNIVERSITY)
by)

Schedule 1

The attached Agreement is drawn up between ______________________________ (“the client”) and Monash University to clarify and formalise the relationship that exists in respect to the project that is being undertaken. The project is a systems development which includes a significant computing component. The University does not charge for this development.

The system is being developed by third year computing students at the University as part of their Bachelor of Computing course. It will form the basis of their assessment for the subject SFT3500/SYS3030 SOFTWARE DEVELOPMENT INDUSTRIAL EXPERIENCE PROJECT.

The development is in fact a simulation of a commercial situation between the students acting as consultants and ______________________________ (“the client”). In this situation the University cannot warrant the system nor can it be liable in the event of the system failing after it has been delivered.

In the course of development, ______________________________ (”the client”) will need to provide specific details of the system. For this reason the University would not sell nor license the system to third parties without the prior approval
of ______________________________ (”the client”). On the other hand
if ______________________________ (”the client”) wishes to commercially exploit the system, arrangements would need to be negotiated with the University.

The ______________________________ (”the client”) rights under the agreement cannot be assigned nor otherwise dealt with without the University’s prior written consent.

Schedule 2

Project Team Membership
Part 1
Supervisor:

Client:

Project:

Part 2
Team Members:

SURNAME	Given Name[s]	Student No.

9 Deed

DEED made this ______________________ day of ______________________ 199 .

WHEREAS

- ______________________________________ has enrolled at Monash University to undertake the subject SFT3500/SYS3030 Software Development Industrial Experience Project for the degree of Bachelor of Computing. The subject involves the design of a system for a Client in a simulated consultancy situation.
- ____ has been appointed as the student's supervisor.
- Under Statute 11.2 of Monash University, a student shall assign ownership of intellectual property to the University if the intellectual property is prescribed under the Intellectual Property Regulations.
- Under Section 2.2 of the Regulations made pursuant to Statute 11.2, the following intellectual property created by a student is prescribed:
  - A patent worthy discovery or invention to which the University has made a specific contribution
  - Intellectual property (other than copyright in the thesis) in respect of which -
    - Background intellectual property owned by the University has been utilised
    - The University has stipulated that the intellectual property must be assigned to the University
    - The University's stipulation has been made prior to the use of the University owned intellectual property
- The University will provide the student with University owned intellectual property in the arrangements for a suitable project to be carried out by the student and the student is required to assign intellectual property arising from the project to the University.

This **DEED** Witnesses:

1. **The Student**
   - Hereby assigns to Monash ownership in any intellectual property arising from the project
   - Agrees to keep secure and confidential and not to disclose to any third party without the written consent of the Client any information concerning the Client's business, including client lists, software, circuit diagrams, hardware specifications, methods, systems, algorithms, and other elements created or developed in the course of the project.
2. **Definitions**

- **Monash** means Monash University a body corporate established by the Monash University Act 1958, and located at Wellington Road, Clayton.
- **The Client** means ____ .
- **Intellectual Property** means any copyright work, circuit layout, eligible layout, design, patent, invention, confidential information, know-how, plant variety and any related right.
- **The project** means the project for assessment in the subject SFT3500/SYS3030 Software Development Industrial Experience Project being undertaken by the student for the degree of Bachelor of Computing at Monash University.
- **The student** means ____ .

In Witness Whereof these presents have been executed a **DEED** the day and year first herein written.

Signed Sealed and Delivered by)
(the Student))
)
____

in the presence of: