Proceedings of SE:E&P'96, Jan 24-28 Dunedin, New Zealand, IEEE CS Press.

Experiences with Facilitating Student Learning in a Group Information Systems Project Course

John C. Grundy

Department of Computer Science, University of Waikato
Private Bag 3105, Hamilton, New Zealand
jgrundy@cs.waikato.ac.nz

Abstract

One of the main aims of university education is to help students become intellectually independent. As the Software Engineering and Information Systems fields are changing so rapidly, such independent thinkers are essential. This paper describes a third-year Information Systems Project course we have designed to facilitate the process of university students becoming real-world software practitioners. The course covers a diverse range of Information Systems Development topics. Students work together in groups throughout the course on a single project chosen from a real-world business client. The organisation and rationale for the course structure are described together with our experiences with the course evolution, its successful and unsuccessful aspects, and student and industry feedback.

1. Introduction

Haigh [13] identifies a key goal of university education as being to help students learn how to take responsibility for aspects of their own learning - to become intellectually independent. He also identifies an important characteristic of many of today's occupations:

"...the exponential growth and change in knowledge and skills associated with many occupations, and the need for people who have the capacity to: readily update their own knowledge and skills; transfer knowledge and skills into new and changeable contexts; and contribute directly to the construction of new ideas and the development of new skills" [13]

This is particularly true of the Software Engineering and Information Systems fields. Because of the rapidly changing nature of software and Information Systems development practitioners must be able to readily update their knowledge and skills, be able to transfer these to other areas, and be proactive in developing new ideas and skills. This includes the ability to: work on large or small projects alone or in groups; manage projects effectively; work and communicate with clients and co-workers; and have an increasing range of diverse technical skills, including those for systems analysis, design, implementation, testing and maintenance.

Many university graduates, however, come out of their educational institution with little idea of what will be required of them in their industry roles. Students often have little experience of "big" computing (i.e. working on large projects) [4], poor written and oral communication skills [5], out-dated or inappropriate technical skills, and little appreciation of the needs of the businesses they are expected to build software systems for.

One of our approaches to addressing some of these problems is by requiring students graduating in the Software Engineering or Information Systems areas to have taken our third year course 0657.315 Information Systems Project (ISP). ISP is currently taught as a third year, B semester course. Students organise themselves into groups of 4 to 6 and choose a client from the local business community. They then analyse the Information Systems needs of their client, design and build a prototype Information System addressing some of their client's IS needs, and present and document their findings in a variety of ways.

A key aim of this course is to let students make many of the important decisions about their project among themselves and in conjunction with their client. Such an approach has been shown to promote "Active Learning" in higher education courses, thereby better facilitating student learning [2]. Weekly meetings with the course lecturer and demonstrator help to keep students' projects "on track" although the students themselves take a large amount of responsibility for the direction and the resulting success or failure of their project work. Project planning and management, peer-assessment, client appraisal, and written and oral presentations are all important aspects of their work in addition to more technical systems analysis, design, prototyping and implementation tasks.

Section 2 briefly puts our ISP course in the context of our wider degree structures and summarises the main content of the course. Section 3 discusses one of the key course aims, which is working in a group for the entire semester. Section 4 discusses another key aspect, working with a real-world industry client. Section 5 describes the course topics in more detail, Section 6 the CASE tools used, and Section 7 assessment strategies. Section 8 discusses our experiences with this course

over the past few years and Section 9 summarises the contributions of this paper.

2. Course Organisation and Evolution

At the University of Waikato the Department of Computer Science is part of the School of Computing and Mathematical Sciences. The School offers as its principal degree the Bachelor of Computing and Mathematical Sciences (BCMS), a four year honours degree. In addition, a three year Bachelor of Science (BSc) degree is an option for those students wanting a more general, non-honours degree.

Both BCMS and BSc students enrol in one of four Computer Science Streams: Information Systems (IS), Software Engineering (SE), Artificial Intelligence (AI) or Computer Technology (CT). Our ISP course is taken in the third year by all students enrolled in the SE and IS streams, as well as by some students in other streams or enrolled for other degrees offered by other Schools of study. In fact, ISP is a very popular course taken by a wide range of students not only doing Computer Science but also Social Sciences, Management Studies and Science and Technology. ISP is seen as a good third year course for gaining a wide range of Computer Science skills.

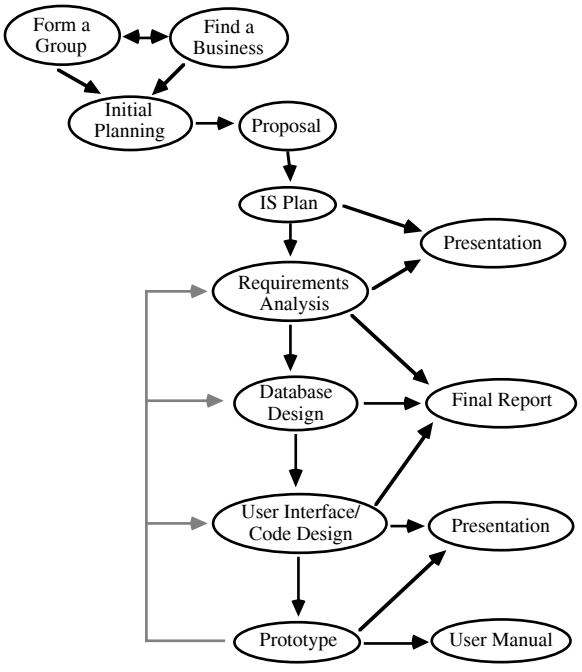

Figure 1. Main ISP stages sequence.

For the IS and SE streams, in year 1 students take the courses Introduction to Computing (IC) and Introduction to Computer Science (ICS). In year 2 they take Systems Analysis and Design (SAD), Programming with Data structures (DS), Program Design (PD), and Computer Organisation (CO). ISP has the SAD and DS courses as its prerequisites. The third year software engineering course (SE) is run in the opposite semester to ISP (SE first semester; ISP second semester). Almost all students taking ISP also take SE. Options in the fourth year include Advanced Database Systems (ADS), Advanced Information Systems (AIS), Advanced Software Engineering (ASE), and Advanced Human-Computer Interaction (AUI).

Figure 1 shows the main stages of an ISP project. Students firstly organise themselves into groups and then find a client from local industry. They then write up a proposal, carry out a detailed Information Systems Plan and write an IS Plan report. A subsystem from the IS Plan is chosen to design and prototype for the rest of the course. Students carry out a Requirements Analysis for the subsystem, design and build a relational database, design a user interface and batch processes, and build a prototype of their system. Results from later stages are used to refine the documentation for earlier stages as necessary. Students write a user manual for their prototype and a Final Report incorporating their Requirements, Database Design and System Design, and documenting their prototype and its installation and maintenance. Students also do two oral presentations; one presenting their IS Plan and Requirements findings and another demonstrating their prototype and describing its design. Client feedback is used extensively for every stage of a project.

The ISP course has evolved greatly over the past several years. Originally it was the third (final) year project course for BSc students before the four year BCMS degree existed. It was then changed to have more management content and no prototype, with each group member working on only some stages. All students are now expected to contribute to each stage of the project and thus gain at least some experience with all aspects of IS development. Recent changes by the author have concentrated on providing additional project management resources and guidance to students, better example projects and general course resources, more extensive CASE and database usage, and modification of the course to suit semesterisation. The most important recent change has been to better facilitate students making decisions about their project direction and critically evaluating their work (discussed further in following sections).

3. Group Work

3.1. Group Formation

One of the key aspects differentiating ISP from most other courses is its emphasis on group work. Students organise themselves into groups of 4-6 members and work in this group for the entire course. Originally group membership was determined randomly by the course lecturer, as in many other group project courses [18]. This was found to have many problems, particularly when students found themselves working with others they had difficultly getting along with. While this helps to teach students about working with colleagues they don't choose in the real world, we found it often had a significantly detrimental effect on the students' learning ability and work quality. When students were allowed to form their own groups, there was a marked increase in productivity, generally less group interaction problems, and students seemed more willing to negotiate about difficulties they had interacting than if their groups had been determined for them.

All of the students taking ISP must have taken the stage 2 Systems Analysis and Design (SAD) course, which involves some group work. Many student groups seem to be carried over from this course to ISP, which has resulted in many groups already having experience in working together and developing effective communication channels. This is not, however, always the case, wih some ISP students are very careful to ensure they do not work with others who they've had bad experiences with in SAD!

3.2. Group Organisation

For each project milestone a member of the group is designated "group leader". The leader's job is to ensure other group members know what tasks they are to be working on, when these tasks are due, and to ensure this work is carried out. The leader acts as first contact point for the course lecturer when discussing progress with the group. The leader is always expected to be able to answer any questions about progress on the current milestone.

In previous years the group leader was chosen at the beginning of the course and not changed. This proved to be fraught with problems. Often it was those group members who were more outgoing or had Management Studies experience/background that were chosen as leader. These choices were often not ideal and much tension between the chosen leader and other group members resulted. In many cases groups decided to change their leader part-way through the course, often causing further friction.

The current approach has been used for the past two years and has proved much more satisfactory. Each member of the group has a turn at being group leader for at least one milestone enabling everyone to gain experience of being in charge of a large group project. Perhaps more importantly, group members find out how difficult it can be to lead such projects, and how important it is for other members to assist the current leader. This usually manifests itself by students being "difficult" before they have had a turn at leadership and then becoming surprisingly helpful after their own experience of leadership!

In addition to a group leader, a member of the group is designated "record keeper" and maintains copies of meeting minutes, task descriptions and allocations, and drafts of deliverables. Jobs such as report compilation and proof-reading, and documentation of designs using CASE tools, word processors and drawing editors are rotated to give everyone experience with these activities.

A key aspect of ISP is that the lecturer does not force group members into particular roles; the group makes these decisions and informs the lecturer of them. The lecturer may comment on particular task allocations and will step in to reorganise a group in exceptional circumstances. Usually, however, we have found students gain much valuable experience in group management by having reasonable freedom to freely debate task allocations themselves.

3.3. Meetings

Groups meet once every odd-numbered week with the course lecturer and once every even-numbered week with the course demonstrator. These meetings are designed to chart a group's progress on their project and to provide a regular forum for questions or discussion with the lecturer/demonstrator. Meetings are usually scheduled for half an hour and groups must prepare an agenda in advance listing the main topics for discussion. This encourages the students to plan their meetings in advance and to come prepared. Each group member will have familiarised themselves with necessary information and each member is expected to be aware of their own and each other's tasks. The group leader is questioned about agenda items with participation by other group members actively encouraged.

Groups must also organise regular meetings among themselves and with their client. Groups usually have 2-3 meetings per week themselves to discuss progress on tasks, allocate or reallocate tasks, and discuss results. Experience with ISP has shown the more organised groups are in planning for meetings (i.e. having proper agendas) and the more frequent and regular their meetings are, the better the resultant project. This is emphasised to students although often groups tell us they only start to fully appreciate the importance of communication and organisation in their group project towards the end of the course.

3.4. Project Management

To facilitate both the student groups and the course lecturer having a clear description of the tasks required and allocated for a particular milestone we have developed several project management resources.

A *requirements checklist* for each milestone is handed out just before the commencement of each milestone and details the main tasks a group must perform for the milestone. It is not an exhaustive list but summarises a minimal set of tasks which must be allocated amongst the group and usually details a minimal set of contents for each deliverable. An example of the requirements checklist for the Information Systems Plan report is shown in Figure 2.

A *task plan* sheet for each milestone is filled out by the group at the start of each milestone and lists the allocated tasks for group members. This serves two purposes: first, to motivate the students into planning their milestone activities before starting their work, and second, it gives the course lecturer a chance to comment on this allocation and to keep a record of the students' initial task allocations. Students often comment it is initially difficult to estimate "difficulty or complexity" of tasks but they become better at this as the course progresses. An example of a filled-out task plan is shown in Figure 3.

0657.315 INFORMATION SYSTEMS PROJECT

CHECKLIST OF ACTIVITIES FOR WEEKS 2-4
(period ending Mon August 7th, 1995)

Information Systems Plan

#	*Activity*	*Progress Summary*

1	Do a scaled-down Information Systems Plan (ISP) for your client organisation (either for the whole or for a significant part of the organisation). Prepare a task plan for the development of your ISP detailing who will prepare each part of the report. Get approval for your plan from the lecturer. Include at least the following in your report: - Statement of organisation's mission & objectives - Executive perspective of general business problems and possible solutions and benefits - Brief description of main business entities - Entity-Relationship diagram of main entities - Brief descriptions of main resources & activities - Relationship matrix of activity vs entity - Analysis and matrix of current computer systems support vs activity, if any (i.e. what have they got?) - Overall Information Architecture: group activities to identify subsystems and the dataflows between them - Identification of IS projects from the subsystems - Implementation Priorities of IS Projects : Technical feasibility : Organisational feasibility : Financial feasibility : Evaluation and Ranking of projects	
2	Develop recommendations and action plans for long term IS development for your client organisation. Include a Gantt chart of projects' schedule.	
3	Discuss with course lecturer to identify the IS project(s) that your group will analyse and design in greater detail for the rest of this course, given the period and resources that you have.	
4	Prepare an ISP report and submit it by Monday, August 7th, 1995 MAKE SURE YOU HAND IN A TASK COMPLETION SCHEDULE ETC. WITH YOUR ISP REPORT.	

Figure 2. Requirements checklist for IS Plan stage.

A *task completion* sheet is the same as a task plan sheet but is filled out when the deliverable is given to the lecturer and client. This specifies who actually did which task and how long/complex the task really was. It greatly aids assessment of individual group members performance to compare the task plan and completion sheets. Often there is quite a difference between the two. This is good as it indicates the group has been proactive about reallocating tasks or helping each other when necessary.

Task monitoring and *task specification* sheets specify more detail about individual tasks and the progress of a group's tasks.

Meeting agenda/minutes sheets serve as templates for the planning of meetings and recording of minutes. Students hand these in with their task completion sheets on delivery of a milestone. These sheets give important additional information which helps the course lecturer assess individual student performance and the performance of the group as a whole on a milestone.

Billable hours sheets record the main tasks carried out on a milestone by each member of the group and how many hours were spent per task. These are useful both for the lecturer (to ensure students are not working too hard/lightly) and for students to demonstrate the amount of time they have spent on their project and how much they could expect to be paid for their work if they did it as contract Systems Analysts.

0657.315 INFORMATION SYSTEMS PROJECT

TASK PLAN

Group: SSS **Approved:** J. Grundy
Date: 10/3/94
Project: ISP

Plan of tasks assigned

For your task plan, indicate the estimated difficulty on the following scale:

5	4	3	2	1
Very complex/ difficult/long	Complex/ difficult/long	Moderate effort	Easy/ short	Very easy/ short

Mem.	Tasks assigned	Diff	Due
Grace	interview client, business goals, activities	4	17/3/94
	activities vs entities, IS architecture	4	21/3/94
	IS criteria	3	29/3/94
	ranking, choose system	3	5/4/94
	group leader	5	-
Sam	interview client, business goals, activities	4	17/3/94
	activities vs entities, IS architecture	4	21/3/94
	IS criteria	3	29/3/94
	ranking, choose system	3	5/4/94
	proof read ISP report	2	10/4/94
Hone	interview client, entities, ER model,	4	17/3/94
	activities vs entities, IS architecture	4	21/3/94
	IS feasibility	4	29/3/94
	ranking, choose system, schedule	4	5/4/94
	compile ISP report	2	7/4/94
Kirsty	interview client, entities, ER model,	4	17/3/94
	activities vs entities, IS architecture	4	21/3/94
	old system (entities, activities, architecture)	5	29/3/94
	ranking, choose system, schedule, appendices	5	5/4/94

Figure 3. A sample Task Plan for the IS Plan stage.

The sheets given to students have proved invaluable for helping them more effectively manage their projects. Students also make use of tools, such as MacProject, but chiefly plan, manage and document their project work via the templates given to them. Students are also referred to extra materials available through the Teaching and Learning Development Unit of the University of Waikato, which provide assistance with developing time management, reading and writing skills [20]. Previously we let students come up with their own techniques for project management but this worked poorly for most groups. In a semesterised project course we believe students should be given reasonable guidance on how to manage their group work effectively.

4. Working with a Real-World Client

One of the more interesting aspects of ISP is that students must find a real-world client to work with from the local community. This is usually a business, but clubs and charities can also be studied. The group meets with their client to determine how the client's organisation works and what the client's Information Systems needs are.

The rationale for using a real-world client and project is to give students experience both with interacting with a "real" client and in dealing with a "big" problem. Many of the case studies and projects used in University Software Engineering courses are rather contrived and usually well-scoped, for practical reasons [21]. This helps students learn the things intended by ensuring chosen examples and projects embody them. Our approach in ISP has been to let students find a real project to study, with all of the advantages and disadvantages this entails.

The advantage of a real-world project is that students must understand the data and functional needs of a real organisation, thus gaining an experience of "big" computing [4]. Students must also interact with a real

client, and thus develop oral and written communication skills for people who may not be computer-literate. Finally, students gain a great sense of satisfaction when they produce a design and prototype for a system which solves the problems of a real organisation.

The main disadvantage of a real-world project is the unpredictability. The structure of some businesses can be strange requiring students to spend a lot of time becoming familiar with them. Clients can have unrealistic expectations from groups, assuming a fully implemented system will be produced. Finally, a chosen business may occasionally undergo a disaster meaning the students can no longer work with them. Two recent examples include a client who was investigated for fraud and subsequently went to jail, and an organisation whose business was taken over and the client a group was interacting with was fired!

Our experiences with ISP have convinced us the disadvantages are much outweighed by great advantages of having students work with real organisations and clients. The course lecturer can, if necessary, take over as the "client" and provide appropriate feedback if an organisation goes out of business or decides to no longer work with the students. This is not ideal, but has worked satisfactorily on the rare occasions it has been necessary. Students often learn a valuable lesson that in the real world, projects seldom go as well as theoretical examples used in lectures.

5. Topics Covered

ISP uses a Modern Systems Development Lifecycle approach to developing a system for a client organisation [9]. The course text, "Systems Analysis and Design with CASE Tools", by Len Fertuck [9], is supplemented by course notes. The notes discuss things to be done to complete each milestone, examples of appropriate task plans, example milestone subtasks, and tutorials for the CASE tools and 4GLs used.

Two lectures are given each week for the semester, and lectures focus on a continuing case study of an "example" ISP group and client. Students thus see the progress of an example project unfold before they attempt each milestone, with the course notes containing extra details for the example milestone.

ISP concentrates primarily on project management issues and the phases in the Modern Systems Development Lifecycle. ISP does not directly cover wider Software Engineering issues, such as testing, version control and configuration management, maintenance, and ethics [1, 8, 19]. These issues are covered in the part 3 Software Engineering course which almost all ISP students also take.

5.1. Proposal

Students initially write a brief proposal document for their client. This provides a brief overview of the group members, the client organisation, their client's main problems, the methodology the group will use, and a simple financial proposal. It also restricts access to confidential information about the client and waives all monetary rewards (as its a student project and their credit is a grade at the end of the course). Students then get approval for their proposal from their client and the lecturer, and sign a Memorandum of Understanding with their client to confirm they will be working with them.

5.2. Information Systems Planning

Students carry out a scaled-down Information Systems Plan for their client organisation. This identifies the important data and information processing activities of the organisation, organises these into IS subsystems and an IS architecture, and simple feasibility studies and criteria ranking are applied to each subsystem. The group normally chooses the top-ranking subsystem which is then refined in the Requirements Analysis phase, and a recommended schedule of development is given to the client.

5.3. Requirements Analysis

The subsystem chosen from the IS Plan is analysed to determine what data and functionality the system should embody. Any current system the client organisation has (whether paper- or computer-based) is analysed. A new system is specified using Entity-Relationship diagrams, Dataflow Diagrams and Process Specifications. A set of non-functional requirements are developed to specify constraints on the new systems operation. A draft subreport is handed in to the lecturer and given to the client for comment. This is combined with subsequent draft reports at the end of the course to form the final System Specification and Design report.

5.4. Database Design

The Entity-Relationship model from the Requirements Analysis phase is extended and a relational schema generated. This schema is normalised and relational database tables implemented using a Relational Database Management System. A draft report is given to the lecturer and client for comment.

5.5. System Design

User interface design includes specifying form and screen formats, report formats, input/output devices (bar code scanners, EFTPOS, etc.), and a physical dataflow diagram capturing the overall flow of control between screens and reports. Batch processes are specified using structure charts and structured English. The Microsoft Access 4GL Form and Report designers are used extensively when designing the interface for the subsystem, in effect forming a rapid prototyping methodology. Students elicit comments from their client and lecturer on their evolving designs.

5.6. Prototype and User Manual

Much of the prototype subsystem has already been developed by this stage. Students create extra forms with command buttons which invoke their forms and reports, to provide an application structure for end-users. Macros are used to provide extra functionality, such as validity checks, calculated fields, buttons for further linking of forms and reports, and an improved interface for forms

(skipping read-only fields etc.). This prototype is demonstrated to the client and lecturer for comment.

5.7. Implementation and Final Report

Students document their prototype with Action Diagrams and Data Navigation Diagrams. These specify where data shown on forms and reports is obtained from, and the tab-order, expressions for calculated values and any constraints of forms. A user manual is written to assist users of the prototype.

Additional information about installation and use of the prototype on the client's computers is described. This includes backup, recovery and archiving policies, integration mechanisms for existing subsystems the client has, and recommendations for further development of the prototype.

The Requirements Analysis, Database Design, System Design and Prototype documentation and implementation policies are then compiled into one final System Specification and Design report. This is printed and bound, and copies given to the client and lecturer.

5.8. Oral and Written Presentations

In addition to developing the computer software and written work described above, students give two oral presentations. At the end of the Requirements Analysis phase each group gives a 30 minute presentation to their client, lecturer, demonstrator and other groups. This presentation outlines the client's organisation, their client's IS needs, their IS Plan findings, and their draft Requirements Analysis findings. At the end of the Prototype construction phase each group gives another 30 minute presentation to the same audience. This briefly describes their Database and System Designs and demonstrates their prototype's functionality.

Several researchers have identified the lack of writing skills taught and practised in many Computer Science and Information Systems courses [5, 14, 22]. ISP is the course with the most written work in our curriculum and thus fulfils a valuable role in allowing students to develop these skills. In particular, students not only practice technical writing when documenting their designs and prototype, but write for their client (proposal and IS Plan report) and end-users (prototype user manual) [6]. ISP also emphasises oral communication skills, both via the two formal oral presentations and, more importantly we believe, by regular meetings between group members, the client and the course lecturer and demonstrator [7, 14].

6. CASE Tool Choice and Usage

Students are encouraged to use a range of tools to assist them in designing, constructing and documenting their systems. MacProject is used to organise projects tasks, to develop PERT and Gantt charts to plan and monitor progress, and to schedule meetings.

The educational version of the Deft CASE tools is used for Entity-Relationship, Dataflow Diagram and data dictionary definition [17]. Most groups use Deft, as it provides a quick way to build diagrams and have their consistency checked. The educational version is somewhat restrictive, however, as it sets limits on the number of diagrams and diagram elements allowed.

Microsoft Access is used by most students to design and prototype their system [16]. Access has proved much better than older systems, such as Ingres, that were previously used. Access provides high-level table, query, form and report builder facilities, in addition to (rather cumbersome) macro facilities. Most groups have found Access to be a very good prototyping tool. Screen dumps from Access also assist with prototype system documentation and user manual preparation. We have been using version 1.0 of Access, which has caused problems due to several bugs.

Students use Microsoft Works for word processing, free-hand diagram drawing, and basic spreadsheet capabilities. Some groups have used PowerPoint for their presentation preparation and delivery.

A problem experienced by most groups is lack of integration betweeen disparate tools, a problem experienced in many organisations using CASE tools [3, 11, 15]. For example, updating a Deft or Access design is not automatically reflected in Works textual documentation. Changing an entity-relationship diagram in Deft is not reflected in the corresponding Access tables. Many groups hand in final reports with inconsistencies between sections due to this lack of tool integration. This has helped to further motivate research work of the author into Software Engineering Environment integration [11, 12, 10].

7. Evaluation and Assessment

Feedback from clients, the course lecturer and demonstrator, and from other students, is elicited for each milestone. Groups can modify their subreport drafts to improve their design's quality. Feedback from clients is particularly welcome, both on the report and prototype, and on the students oral presentations.

Figure 4 shows the relative marks for each deliverable in the ISP course. The group work component of the course makes up 65% of an individual students final grade. The individual mark is 35%, made up from the mid-semester test and a "miscellaneous" mark given to each student. This is based on a range of criteria, including the student's performance in meetings, the complexity and duration of tasks they have worked on, and how well they have managed the project and assisted with managing the project.

Milestone		% Contribution
Reports		
• Proposal		5
• Information Systems Plan		10
• System specification and design, which comprises the following:		
	• Requirements analysis	5
	• Database design	5
	• System Design (user interface and code)	10
Prototype	complexity, quality and user manual	20
Presentations	2 presentations on progress-to-date	10
Test	1 test	15
Miscellaneous	Other marks given to individual students: • performance in group meetings • meeting and presentation attendance • complexity of tasks performed • group management and organisation	20

Figure 4. Assessment in the ISP course.

The group work portion of a student's grade is the same for all members of a group. We recently tried to give different marks per student based on amount and quality of work done for each milestone. While this worked well for the groups comprised of more able students, groups with weaker students had large differences between members' grades. While students didn't complain about this too much, the system was difficult to administer. We now use only the individual mark (test and miscellaneous) to differentiate between students, with all members of a group getting the same mark for their group work. This means a student who has done very well will have a different final grade to one of their group members who hasn't done so well individually, but the difference will not be great.

8. Discussion

Our ISP course has been very successful. Feedback on the course from both students and clients has always been very positive, with the course earning very high Course Appraisal ratings from students. Clients have been extremely pleased with the final reports and prototypes. Most say their business can readily use the results of the project work, and some have employed students to put their designs into practice.

We believe the main advantages of our ISP course have been the use of a real-world project for students to study, having students work on this large project in the same group for the entire course, and having students cover a wide range of Information Systems Development methodology steps. Students also use a range of tools for planning, designing, building and documenting their system. Informal student feedback has always been positive, with many students saying ISP is one of the most valuable courses they have taken.

ISP has proved to be an enjoyable course to teach. The lecturer gets to know each student individually, which is quite different to most other undergraduate courses. Students also enjoy this aspect of the course.

Many students feel the course requires significantly more work than others. We feel this is a fair reflection of the nature of the coursework, with many meetings and organisational activities required in addition to the production of deliverables. It should be noted that those groups who are well organised invariably find the course not to be a heavy workload; it is those groups who are disorganised or communicate poorly who complain of excessive work pressures.

The course has proved to be demanding of the lecturer's time. This is due to both the large number of student contact hours per week (roughly 7-8 hours per week, with 80 students enrolled for the course and split into 15-16 groups), and informal contact hours when students come to ask for advice. Recent semesterisation has reduced this load somewhat, with more demonstrator resources available. The course lecturer must, however, be prepared to spend a good deal of time working with groups and individual students to ensure the projects, and the course as a whole, continues to be a success.

Future developments planned for ISP involve a move from structured analysis and design to object-oriented analysis and design. We hope to use Integrated CASE tools to improve consistency between different aspects of project data. ISP may be run in the first semester of the year, and Software Engineering (SE) in the second. This will give students taking SE experience working on a large real-world project and working in a group, before more general SE topics are introduced.

9. Summary

We have described our approach to giving students real-world group project experience in a year 3 Information Systems Project course. Students work in a group of 4-6 all semester on a project chosen from a real-world organisation. Students examine the IS needs of this organisation, and design and build a prototype system using Modern Systems Development Lifecycle steps. Our course has been a great success in producing students with skills and experience in project management, oral and written presentation, and the designing and building of a complex Information System to solve a real-world problem. Student and industry feedback has been very favourable. This feedback, and the high standard of project deliverables, indicates students finish the course well-equipped for their role in the rapidly changing world of software and Information Systems development.

Acknowledgments

The author gratefully acknowledges the efforts of previous teachers of the ISP course, Dr Zaidah Rizak, Dr Bob McQueen and Mr Bill Rogers, and course demonstrators, Mr Phil Treweek, Ms Angela Martin and Ms Donna Neal. Much thanks go to the industry clients and students who have made the course possible, and their useful feedback in continuing its development.

References

[1] ACM Press, Report of the ACM/IEEE-Computer Society Joint Curriculum Task Force, 1991.

[2] Beard, R., *Teaching and Learning in Higher Education*, Third Edition. Harmondsworth: Penguin, 1976.

[3] Bounab, M. and Godart, C., "A Federated Approach to Tool Integration," in *Procs of CAiSE'95,* Finland, June 1995, LNCS 932, Springer-Verlag, pp. 269-282.

[4] Cohen, R.F. and Menzies, T., "Providing Software Engineering Students with an Experience in 'Big Computing'," in *Proceedings of the 1994 Software Engineering Education Conference,* Purvis, M. Ed, IEEE CS Press, Dunedin, New Zealand, November 21-24 1994, pp. 71-77.

[5] Cunningham, S.J., "Learning to Write and Writing to Learn: Integrating Communication Skills into the Computing Curriculum," in *Proceedings of the 1994 Software Engineering Education Conference,* Purvis, M. Ed, IEEE CS Press, Dunedin, New Zealand, November 21-24 1994, pp. 306-312.

[6] Day, R.A., *How to write and publish a scientific paper*, Fourth. Oryx Press, 1992.

[7] Dumont,R. and Lannon, J., *Business Communication*, 3rd Ed. New York: Scott, Foresman, 1990.

[8] Ford, G., "1990 SEI Report on Undergraduate Software Engineering Education," SEI, Carnegie Mellon University, SEI-90-TR-3, 1994.

[9] Furtuck, L., *Systems Analysis and Design with CASE Tools*. Wm. C. Brown, 1992.

[10] Grundy, J.C., Hosking, J.G., Fenwick, S., and Mugridge, W.B., Connecting the pieces, Chapter 11 in *Visual Object-Oriented Programming*. Manning/ Prentice-Hall, 1995.

[11] Grundy, J.C. and Venable, J.R., "Providing Integrated Support for Multiple Development Notations," in *Procs of CAiSE'95,* Finland, June 1995, LNCS 932, Springer-Verlag, pp. 254-268.

[12] Grundy, J.C., Mugridge, W.B., Hosking, J.G., and Amor, R., "Support for Collaborative, Integrated Software Development," in *Procs of the 7th Conference on Software Engineering Environments,* Netherlands, April 1995, IEEE CS Press, pp. 84-94.

[13] Haigh, N., "Promoting Intellectual Independence: A Legislative Catalyst," June 1994, Teaching and Learning Development Unit, University of Waikato.

[14] Meinke, J.G., "Augmenting a Software Engineering project course with oral and wiritten communications," *SIGCSE Bulletin*, vol. 19, no. 1, 238-243, January 1987.

[15] Meyers, S., "Difficulties in Integrating Multiview Editing Environments," *IEEE Software*, vol. 8, no. 1, 49-57, January 1991.

[16] *Microsoft Access 1.0 User Guide*, Microsoft Corporation, 1992.

[17] Morgan, A., *The Deft CASE Tool*, Department of Computer Science, Massey University, Palmeston North, New Zealand, 1991.

[18] Oudshoorn, M.J. and Maciunas, K.J., "Experience with a Project-based Approach to Teaching Software Engineering," in *Proceedings of the 1994 Software Engineering Education Conference,* Purvis, M. Ed, IEEE CS Press, Dunedin, New Zealand, November 21-24 1994, pp. 220-225.

[19] Summerville, I., *Software Engineering*, Fourth Ed. Addison-Wesley, 1992.

[20] Learning Assistance Guides: Time Management, Writing I & II, Reading, Teaching and Learning Development Unit, University of Waikato, 1994.

[21] Thomas, R., Semeczko, , Morarji, H., and Mohay, G., "Core Software Engineering Subjects: A Case Study ('86-'94)," in *Proceedings of the 1994 Software Engineering Education Conference,* Purvis, M. Ed, IEEE CS Press, Dunedin, New Zealand, November 21-24 1994, pp. 24-32.

[22] Vemurie, V.R., "Computer Science and Engineering Curricula," *IEEE Transactions on Education*, vol. 36, no. 1, 108-110, January 1993.