Enterprise skills: the final frontier !

Mike Holcombe and Helen Parker,
Department of Computer Science,
University of Sheffield.

The emphasis, so far, of what we have been doing in projects like the Software Hut [1], and the Maxi [2], has been on developing skills which are associated with the successful construction of a software solution for a real commercial client. As we have seen these can be categorised as individual personal skills (presentational and communicative, planning and reflective, researching etc.); technical skills to do with software development technology and teamwork skills (working with people, listening, discussing and agreeing, monitoring, leading and delivering).

This is not enough, however. The challenge is to set these activities into a *real* business context, providing an opportunity to market, negotiate, select and cultivate ones customers, to consider the financial aspects of a contract and to take responsibility, generally, for running a successful business. These activities involve what we are calling *enterprise* skills.

1 Identifying the enterprise skills.

The previous chapters have considered how many important personal and teamwork skills can be developed through real project work. We have seen that the fundamental processes relating to successful software engineering are all brought to life in a dynamic and meaningful way through the experiences gained in a real project. These processes include:

requirements elicitation,
problem identification,
business process analysis,
solution scoping,
software specification,
validation,
design,
analysis,
implementation,
testing,
delivery,
maintenance
and system evolution

However, there is more to a successful software engineering education than just this. Many employers complain that graduates have no understanding of the world of business and this is regarded as a serious weakness in the educational process.

> CSSA quote: *Graduates have a lack of understanding of business issues and how to develop business systems. Graduates have technical skills, but are unaware of how they relate to industry.*

A key challenge for educators is to try to see how these can be developed in our senior students within the degree programme. One way forward, as we have discussed, is the idea of a placement or internship in a commercial organisation for a few months or a complete year. On the surface, this has its attractions. However, we must consider the benefits more carefully. Placements vary enormously, many are very rewarding but others involve some very routine types of activity which are little more than cheap labour. Monitoring the quality of these activities is almost impossible.

We need to ask to what extent is it likely that a placement will provide an opportunity for a student to develop a suitable understanding of business? In a large organisation, perhaps where the student is working in a team insulated from clients and customers, it is probably difficult to really involve the student in many of the core business activities. In a small organisation exposure to these aspects of business may be more likely. Equally, however, the amount of serious training available to help the student may be small in such an organisation.

There is a further issue, however. An entrepreneur has to consider how to take responsibility for the development of the business, taking key decisions that will affect its future success. Such opportunities are likely to be rare in a placement but it is not until these are faced that the most basic of business issues will be addressed.

Improving students job prospects and employer satisfaction is not just about developing communication and teamwork skills or about having attended modules in business studies or accountancy, it is much more fundamental than that. The belief is that graduates do not understand the context, the processes and the constraints that apply to business these days. As such they are poorly prepared for the challenges that the dynamic world of work will pose.

The future employers of students with *enterprise skills* will find a much greater understanding of the reality of business among their recruits and so will not have to expend precious resources on dealing with the problems that can arise from the normal lack of such understanding.

It is in response to the challenges of trying to introduce the entrepreneurial dimension that the concept of a student run software company evolved as part of the curriculum. This chapter looks at the organisation and success of this activity over the last three years.

2 The Genesys experiment

Fourth year M.Eng. students undertake a large exercise, amounting to one third of their total studies, running a software house. The student run company is called Genesys Solutions (the first cohort of students called it VICI, this was changed to Genesys by the second cohort and the name has become established, principally since it now has some marketing credibility and has received much favourable publicity, locally and nationally.). Currently, it involves involve 21 M.Eng. and MSc students.

This company is an attempt to take the process of students working for *real clients* to its logical conclusion, as far as an academic venture can go. The students run their own company, organise themselves into teams with quality control, network administration and archivist roles all clearly identified within the company. The project runs over two semesters and counts for one third of their total effort in their final year (of the M.Eng. degree) i.e. approximately 400 hours per student over the year. They meet in a weekly board meeting, review progress with their contracts and decide on future business activities.

We, as their lecturers, attend in order to provide advice where necessary but leave the decisions to the board. The company has its own laboratory and network of computers operated independently of the department’s laboratories, paid for from company earnings.

Each month the company produces a detailed report on its business activities, detailing the design and testing status of the various projects. In the past year the company has worked with nine external organisations with each student involved in 4 or 5 different projects during the year. They spend a lot of time negotiating and dealing with their clients and this is invaluable experience. Because they have almost complete control and responsibility for their company their motivation is exceptional and the quality of their work is very high. We believe that this project is unique.

2.1 The history and evolution of Genesys

The 4th year M.Eng and M.Comp are prestige undergraduate degrees, with high entry requirements and a hurdle at the end of the 2nd year that must be satisfied, of an average mark of at least 60%. It is also the main route that we perceive to CEng accreditation following the requirements of SARTOR [proj98.doc 2]. So the class is fairly small, talented and enthusiastic. In addition to these credentials the students are required to have participated in the Software Hut project in their second year (see Chapter Y)and to have completed a third year individual project.

When we first started exploring the idea of setting up such a course we talked to the students - then in their 3rd year. Their response was very enthusiastic and throughout the planning period we worked with them and obtained their support for the basic structure and curriculum details. This was important because the assessment was going to be difficult to define precisely in advance and a spirit of trust was needed between staff and students.

The emphasis of the work is on learning how small IT companies are created and managed, the legal and financial frameworks with which such companies operate, the practical management of the companies and their successful trading.

Students involve themselves in the following activities:

Researching market opportunities for software products,
Carrying out IT audits on behalf of local organisations and preparing appropriate IT strategies,
Acting as software/computing consultants to local organisations,
Developing software for clients,
Maintaining software for clients, possibly including maintenance of former student projects.
Delivering training courses.

Unlike the Software Hut, the students on the Genesys course have full responsibility for negotiating possible projects with their clients. We provide contacts where necessary and the students must follow these up and eventually to decide whether to proceed with a project and under what terms and price. We intend that, as far as possible, the students should be running a fully independent company. The University provides the legal and financial accounting framework but the students take responsibility for supplying all of the decision making, documentation and management information needed for a real, independent company.

2.2 Genesys now:

The aim of the Genesys course is to give the students experience in business process modelling, requirements capture and all of the processes involved in the engineering of a piece of high quality software for their clients.

This will also include other aspects of company activity such as organising and delivering training courses. The training courses they will be providing for an industrial client have to be delivered to a high standard. The client satisfaction will be partly measured by asking the participants of the course to evaluate the quality of the teaching. They are now on the other side of the fence from when they evaluated the teaching of the university lecturers!

The Genesys course runs throughout the academic year and students are often willing to work on after their exams so that each year runs into the next, so one potential benefit of the Genesys course is that the companies might provide some maintenance services to organisations that have provided student projects in previous years, perhaps as Software Hut or MSc projects. This is something that we would otherwise not have been able to offer and which has, in the past, proved a disincentive to involvement with student projects for some potential clients.

This may prove to be useful in securing further work for this and other group projects since the CSSA describes after sales support as one of three ‘deal clinching’ aspects of an IT supplier’s business. It also brings home to each cohort the need to ensure a high quality product so as to minimise the amount of perfective maintenance the company will have to do in following years.

2.3 Company organisation and infrastructure.

During the course of the Genesys module the students keep company records and prepare company reports as well as developing analysis and design documents and other consultancy reports for clients.

In the first two years of the course the students attended a small business training course run by the local TEC once a week for the first six weeks. The aim of this was for them to learn the basics of bookkeeping, business law and business management. These have since been replaced by more focused sessions on negotiation skills and other topics as and when needed. The reason for this is that the training should be based on the needs of the students (demand led) rather than by the availability of standard courses (supply led). This has worked better but it does mean that the lecturers must monitor activities critically and discuss the training needs of the company members on a regular basis.

The way that the students have organised themselves into teams has evolved during the development of Genesys. Currently each team comprises 4 students, they each have a client portfolio and are involved in one development project at a time. Each team also acts as independent reviewers and testers for another teams projects. This has increased quality considerably. Every deliverable, which includes all documentation and all software has to be independently checked. This applies to all versions of all deliverables. The rule of thumb used is that testing in general will take as much effort as development. Two students also take the role of systems administrators, taking full responsibility for the support of the company infrastructure. These students produce monthly records of the system and archive and backup all of the system. These systems administrators have a slightly lower load in the development projects as a result.

A number of management planning systems must be set up by the students that involve methods of identifying the work involved, the way it is to be shared out and some form of monitoring system which records effort, and tracks activities and deliverables. The students also have to think about quality control methods, should they use inspections, some formal mechanism (Fagan, for example, or use some of Gilb's ideas.) or what? They must also think about some way of costing activities, some of this will be with real money, but certain clients, particularly voluntary and charitable organisations, will not be charged so there is a need to estimate the cost and account for it in a "virtual" way in the company accounts. The attitude of the lecturers is to talk to the group regularly about these matters, but not to prescribe in detail what they should do.

In each month the company provides a monthly report on activities. This includes the current status of each contract, including a commentary on milestones reached, problems encountered and potential solutions. Initially monthly accounts were also presented in two forms - virtual accounts where their labour costs are included and real accounts which reflect real expenditure and income.

The company initially kept virtual accounts (which include notional labour costs based on the number of hours work carried out on the projects and management activities and costed at an appropriate level, for the sake of argument work can be costed at something like £10 per hour per person) on a monthly basis. Some projects might include a notional charge to the client based on an estimate of the time required to complete the project. In subsequent years this practice was abandoned since there were very few occasions when real payments arrived, the normal procedure being payment after delivery. This was a cause of frustration for the students who could see the labour costs mounting but no income arrived until very late in the year!

The report also contains any deliverables completed in the previous month, requirements documents, design documents, implementations, quality control material, user manuals, systems administrators reports etc.

The precise structure of these reports is determined by the students in consultation with the lecturers. The production of these reports demands that detailed minutes and actions be recorded.

Teaching is done by means of lectures, seminars, practical design work and client presentations. Assessment is by coursework. The lecturers have to monitor the progress through weekly meetings with the company, helping with the planning of the work, ensuring that all the students are contributing suitably, reading the monthly reports and meeting the occasional client.

To provide a focus and a basic infrastructure for the student companies the Department has

arranged for a small room to be allocated for their exclusive use, this is furnished with 10 PCs and other basic equipment such as a phone, kettle etc.
The installation and maintenance of any and all hardware, software and records is entirely the responsibility of the students (normally the system administrators): no significant help from the lecturers or University support staff is provided, although maintaining the good will of the technical staff is important to the smooth running of the company .

At the end of the course a final report is produced composed of the monthly reports together with assessments of the state of the various projects. Any plans for continuing uncompleted work beyond the end of the course, either by the current students or by next year’s group, are detailed if such continuation is deemed necessary and feasible.

In addition the students are each expected to produce a short personal report assessing the progress of the company and the contributions made by themselves and the other members of the company.

Copies of all documentation produced for the clients are also included in the report.

3 Experiences from the Genesys project.

Although the Genesys project has only run for four cohorts of students there are already many conclusions that can be drawn from the experience. It is certainly nothing like any of us have ever done before, whether staff or student. This, naturally, brings with it some problems. The company has evolved over this time as we leant from our mistakes and faced new challenges.

The overriding impression is, however, one of exhilaration, both for students and staff, in really getting to grips with the most important issues of the modern software enterprise business. We consider some of these benefits for the different stakeholders next.

3.1. Developing a professional attitude.

To their surprise the students learned that a professional attitude and image is one of the most important assets a business can have. The students were pleasantly amazed by the effect on clients attitudes of wearing a suit. They were equally impressed by the enormous improvement in the quality and volume of their own work that was brought about by the introduction of simple business practices, such as chaired meetings and taking accurate minutes, that they had at first thought were just a waste of valuable working time. The students felt regret that they had been slow to learn this lesson because it meant they had to work harder to catch up and it had antagonised their clients. They did, however, recognise that this delay had taught them a valuable, if painful, lesson.

The future employers of Genesys students will be taking on fresh, keen young graduates but will not have to spend time and effort acclimatising them to professional attitudes and approaches because the students have learned the value of these things at first hand and have for the most part been disabused of the idea that they know better. In addition they will be receiving workers with some understanding of the realities of business, which is an attribute that the CSSA highlights as being lacking in graduates [CSSA.doc]. This understanding may help the students to understand the decisions and attitudes of their superiors, thus reducing friction and increasing the scope for co-operation between employer and employee.

Team work

In all of the groups there has been a strong sense of team spirit within the group with the members generally taking equal credit and blame for all successes and failures. The groups recognise that a general lack of cohesion and discipline in the early stages has led to poor performance of some of their tasks, but by the end, nearly all were convinced of the importance of working in a well organised and cohesive group with clearly defined roles. It is inevitable that some individuals will never be able to fully integrate themselves into a functioning group. The experience with Genesys has been that the numbers are much smaller than in traditional group projects. The more realistic the project setting the less of a problem “loners” seem to be. Since we claim that Genesys is as close to a real software house that we can realistically get in a university context we should expect that a much stronger group ethos would develop than in any other type of groupwork activity. This does seem to have been the case.

Only one member of this group was unable to fit into the team and tended to work alone, ignoring procedure and dismissing the idea that his work might need to be tested or that he should occasionally produce stable versions for the team to look at. However, the group felt that this member’s hard work and natural ability had offset any problems caused by the duplicated effort that his attitude produced. The other members considered that he had made as much of a contribution to the group effort as any other member. In addition the students commented that dealing with this member had given them some management practice and his comments in meetings had been quite entertaining which made for a good working atmosphere.

3.2. Genesys: the benefits for staff and for clients.

The staff supervising the course also gain. Being involved with many types of real problem solving; seeing how varied and volatile the business context is; studying how software development teams operate in a realistic setting and exploring the strengths and weaknesses of current design methodologies have all proved valuable. It is an excellent opportunity to see how relevant academic software engineering is to solving business problems. In the process, we also learn a lot about modern technology and its uptake in local companies.

The University also benefits from excellent public relations. The local business community sees that the "ivory tower" can provide a practical and effective source of expertise for many local organisations, both large and small.

The clients obviously get a very cost effective software solution or low risk prototype and consultancy. They also learn a lot about their company as well as the role of being an effective client. This is an important skill which is vital for a successful project. Typical comments from clients are:

> *I learnt a lot about the structure of my own business from discussions with the students*; *Our company obtained three good quality prototypes, each of which had parts that could be used in a final version of the system, which was brilliant for us. A company can get so much out of a project if they approach it in the right way*; *The overall ability of the students seemed very high I was pleased with what I saw.*

4 Practical operation of an enterprise project.

Running the Genesys project over the last 3 years has taught us a lot about how such projects might be managed. Naturally, in other contexts, there might be better alternatives and we must also continually reflect on the processes so that we can make it better.

4.1. Managing the project.

The important difference of this type of activity from normal (internal) software engineering group projects are the following.

There is an important negotiation phase where the team discusses a possible project with a potential client. There has to be a period of discussion and research during which the client learns more about the company and its track record and the team learns about the client’s business and possible project scope.

Once negotiation has progressed to a certain point both the client and the team have to decide whether to proceed with the project. By this time the team has an estimate of the effort required to build the solution, the technical feasibility of it, and the extent to which the client will pay for the solution. The client has understood more of what might be involved in their organisation such as equipment and training needs.

The next phase is the detailed requirements elicitation. This is always a difficult process and it varies enormously with the character and experience of the client. At the end of this there should be a requirements document agreed with the client. This is rarely the end of the story, however.

One the specification and design has started we are on more familiar territory. However, we insist that thought is given to the quality assurance dimension and that a review and testing strategy is identified.
In the design and build phase there will be many prototypes and pilot systems which will need to be evaluated both internally and with the client.

The final phase of testing, debugging and delivery is potentially a long process, which can often frustrate both client and the team. It is here where the benefits of a clear and stable requirements specification and a rigorous review and testing strategy will pay enormous dividends.

Client Negotiation.
We arranged for the class to have a session of training on negotiation techniques. This was reasonably successful. It raised the issue of preparation for a client negotiation, of selecting strategies, or trying to look at the situation from the client's point of view and of ways of managing the process in a subtle and successful way. Language, both verbal and body language can be important here. During this phase try to decide if the project is feasible, will it be possible to complete it within a reasonable amount of time (always difficult to estimate), do you have the skills, the knowledge, the design tools for the job (learning a new technology brings with it an extra cost), is the client really committed and prepared to give up the time needed for a successful project completion?

Requirements capture.
There are no hard and fast rules here except the following:

Do not confuse the client with jargon or unnecessary detail.
Listen to what they say about their business.
Write everything down.
Be patient with the client.
Do not force them into a specific implementation direction prematurely.
Offer alternatives wherever possible, this often encourages a decision.

Quality assurance.
One of the key lessons learnt with a real client is the need for the highest quality product. It is a disaster if the delivered software is full of bugs or fails to meet the client's needs and expectations. This is a recipe for continual visits to fix problems, re-engineer code and generally spend a lot of time doing what should have been done properly earlier. This means that new projects cannot b started properly and thus new earnings cannot be obtained when expected.

A rigorous quality assurance process is thus vital. It has been well established that a programmer is rarely successful at testing their own code properly. Furthermore, every type of output has to be reviewed or tested. It is no good just focussing on the code if the original requirements document is flawed. Hence we must institute a process where everything is reviewed independently and rigorously. This means that the company procedures and management must be oriented around a quality assurance framework.

We have found that the best approach is to have the company split up into a number of separate teams. These have comprised as few as 2 members but typically we have 4 or 5 in a team. Each team has two roles, they have their own development projects and clients, they also act a quality assurance reviewers and testers for another teams projects. Everything that is produced, documents, designs, codes and test reports is reviewed and checked, with a check list document providing the results and evidence of the review process. Initially this was regarded as a tedious chore but the rapid discovery of errors and flaws in every kind of output uncovered by the process soon convinces everyone of the wisdom of the strategy. An important psychological issue is to encourage the reviewing process to be carried out in a positive and constructive manner. This approach of everyone being involved in both the creative system building process and the possibly destructive critical review process ensures that team members do not take a negative and destabilising approach to testing, after all their own development work is also under scrutiny from another review team.

4.2 Assessment.

Like any experimental course it is necessary to think very hard about the way the assessment will be carried out. The basis of the assessment should be the aims of the course and the outputs from the course activity. With this in mind we developed the following structure for the assessment. Our experiences from the other industrial project modules, such as the Software Hut, was also very important.

(i). Customer satisfaction - this is measured on the basis of questionnaires given to clients on delivery of the product/course and on our discussions with them.

(ii). Administrative procedures - this is based on the conduct of the weekly board meetings, the monthly reports, planning documents, business plans etc. We will consider the way that these procedures are carried out as well as the quality of what is produced.

(iii). Quality control - this looks for evidence of a review process, the use of software engineering methods in specification, design and validation (testing). An important element will be the standard of the technical documents.

Initially we thought about a further aspect, *profitability* - based on the annual virtual accounts - would we have survived financially if the money had been "for real", also the business plans would be assessed with this in mind. However, the difficulty of compiling meaningful accounts when the client base consisted on charitable organisations with little avaliable money as well as commercial companies who paid at almost commercial rates led us to drop this aspect.

We need to apportion the assessment between these (and possibly other) headings. Within these headings we will need to come to some more detailed division.

Because the students have approached the activities with a great deal of enthusiasm and effort it is to be expected that this will be reflected in their overall grade for the module. This seems to be a common phenomenon with a self selected group of students doing project work.

Detailed outputs used for assessment.
Each month the company provides a monthly report on activities. This includes the current status of each contract, including a commentary on milestones reached, problems encountered and potential solutions. The monthly accounts are also presented in two forms - virtual accounts where their labour costs are included and real accounts which reflect real expenditure and income. For several months both of these accounts were somewhat negative since no income had come in as no contracts had been completed. This can be a little dispiriting at first but has now been accepted by the students.

Another assessment source is the documentation associated with each contract. We expect *best practice* in software development to be used, wherever possible.

The monthly reports to us were also of great value and will be part of the assessed material. These reports were structured as follows (a structure that the students refined themselves in consultation with us):
1. Quality checklist (see appendix A);
2. Preface;
3. Development projects - current status (Introduction, schedule from previous month, achievements of current month, problems and processes, schedule for further work, deliverables for this month);
4. Consultancy projects (same structure);
5. Training projects (same structure);
6. Administration (same structure);
7. Appendices (correspondence with clients,
8. Accounts (real and virtual);
9. Minute of business meetings.

5. Evaluation.

A project like this needs to be carefully evaluated if we are to persuade others to join us in this educational adventure. We have tried to determine what the initial expectations of the students were when starting the programme and how these compared with their feelings at the end. We designed questionnaires which gave us some information. We have contacted former students on the course and tried to find out what they think after the event. We spoke to former clients and received some useful feedback. One interesting phenomenon was that when the students went to interviews for industrial jobs this was almost the only thing prospective employers wanted to see.

Ultimately the proof of the pudding is in the eating. Two of last years class have joined with a couple of other postgraduate students in setting up their own company, specialising in internet applications. It will be interesting to see how they fare.

5.1. Educational evaluation.

This is potentially difficult since the main benefits will probably come to light years after the students have left. However it has been possible to see that the students have obtained a much more mature understanding of the business environment and achieved a more responsible attitude to their own work and the discipline of working together in a group. because they have to take so much more control of the projects they have developed a much more sophisticated understanding of the way a group has to communicate, plan and deliver a successful development project.

The academic theories of software engineering have also been tested out for real. The pressures of having a real client and of needing to optimise ones resources to the best effect test these theories to the full. The heavy overheads associated with textbook software engineering may not be worth the effort. However, economising on quality review and testing is definitely foolhardy. This is perhaps the single most interesting conclusion of the class.

The lecturers also benefit. We see how the textbook ideas apply to real projects and can learn many things, particularly about the problems of having highly dynamic requirements. Trying to manage change and the control of versions of documents, software etc. is vital. Much more important than the techniques, tools and notations used.

5.2. Genesys: the student's view of their experience.

Whatever we might think, the student view is likely to be different. The student's expectations about the course when they begin it will colour their attitudes. A number do not realise how unusual such a course as this is, the course concept seemed to be such a natural one to the class.

The following comments are from recent Genesys students recorded by an independent evaluator (in May 1999). Responses by the course lecturers are in *italics*.

Course structure.
Many students commented that there is not enough time to do the work, there are too many other assignments. “For the first two months you don’t do anything, then just as you start to get going on the project and clients are starting to demand action you hit the other assignments. The course should be worth 30 credits per semester, or possibly be done full time over the summer.”

One student has at times been working 38 hours per week just on Genesys and the others do not see this as unusual or surprising.

One suggestion was that the Genesys course should start as a 10 credit course at the end of the 3rd year with the third year students acting as subordinate staff to the M Eng. Students. This would give students time to make all of the mistakes and to learn the ropes without delaying work and antagonising clients and would allow last years’ students to train the next lot.

There is always a problem when courses like this run alongside traditional computer science or software engineering courses. We have noted the same comments with Software Hut and similar projects which involve a real client. Students prefer to work on these projects because they are seen to be more realistic and the client wants their work – it is not going to disappear into a filing cabinet, or worse still, into the bin once the assessment has been carried out! We have extended the credit weighting of the project but that did not cure the problem since whatever weight it is accorded students will be tempted to spend more time on it than on their other work. Exhortation to manage their time in a more balanced way is the only tactic we have been able to use.

The lack of teaching was considered a major drawback especially in the areas of business skills (particularly negotiation) and databases, which form the core of Genesys work. Particular skills that could be taught include: how to negotiate, interview and extract requirements. Also needed are lectures on how to produce feasibility studies, technical reports, testing documents and presentations. The group also felt that lectures on Visual Basic and Access would have been useful.

This year we arranged for an outside specialist in negotiation skills to come in and give such a course. The student response was rather mixed, it was of some help in their actual client negotiations but did not produce a “magic bullet” for them! The courses on the production of documentation, testing and presentational issues were all the subject of 1st and 2nd year courses. It’s surprising how soon these are forgotten!

The students felt that the external business course was completely useless, largely because it was about accounts and marketing, areas that do not have a major impact in the Genesys organisation.

In the first two years an outside course on setting up a business was used. Because of the student response this has been abandoned and we are taking the view that any instruction should be stimulated by need. In other words, if information on a specific topic is needed then it should be the stimulus for a short specialist course. Previous cohorts have not found the traditional academic management courses very useful or convenient (a semester course might not cover the topics of interest until too late).

Infrastructure.

The rapid increase in the numbers of students doing the course caused some problems with a shortage of machines and cramped laboratory.

A new room has now been found and more up to date equipment obtained.

Management and process.

Some students thought that the company needs an MD, but he/she should not be one of the students as this would lead to personal tensions within the group. It was felt that somebody who can give and take marks would be most effective in this role. (One suggestion was a postgraduate co-ordinator.)

To a certain extent this goes against the philosophy of the course. It should be the students' company and the lecturers give advice, sometimes rather critical, but the concept of a line manager is not one we want to develop in this context. We have introduced a postgraduate technical adviser to help in some aspects of the activity.

It was realised by the students that there must be a documents manager and a network manager (posts possibly taken in rotation). It was felt that all duties must be assigned on an individual basis, rather than leaving things to be based round the concept of group responsibility which usually does not work because nobody does anything.

Every year these posts are allocated and the jobs are effectively carried out.

It was felt that presentation was importane. Putting forward a smart, professional image right from the start was vital.

Some thought that we <u>must</u> make money. The projects set up by the supervisors, particularly internal University ones, should not be accepted automatically as they frequently do not pay well.

The group thought that everyone must be convinced early on of the need to follow strict procedures and to allocate responsibility for all activities right from the start.

These include some rather individual comments. In general the issue of making money is not a major one, funds for maintaining the equipment base is the most important issue. In general, the educational benefits of the programme are extraordinary and all the students subscribe to this.

Individual suggestions and comments by members of Genesys:

The students seem to be in broad agreement on most points and so common points previously attributed to another member of the team have not been repeated.

*Rupesh***.**
Initially sceptical about working with people he had never met.

The group broke down into pairs of people who knew each other already.

Gave precedence to Genesys work because it was worth 40 credits and it was ‘real’.

Took on the CRC database thinking it would be a quick bug fix, but discovered that apparently superficial problems had deep causes. This caused some disappointment because he spent all of his time fixing someone else’s mistakes and the project never felt like it ‘belonged’ to his team. It is better to have a project from start to finish.

He suggests that it is very important to ensure that there is a requirements document set in stone and signed off by the client. This project had two clients, one local and one national.

Decisions within the Genesys team were largely democratic and there was broad agreement on most topics.

Testing teams were a good idea.”

*Ben***.**
The network administration he did will be useful for employment.

Learned that requisitioning equipment in a large organisation such as a university can cause major problems for scheduled work and is radically different to the situation of a private individual just going out and buying one.

Learned that trivial technical changes can be enormously complicated on a wider scale.

Learned from a poor first attempt at requirements capture that it is important not to make any assumptions.

Found that he had two clients: one technical and one management.
Also found that talking to a middleman is counter productive and that it is necessary to talk direct to the users.

Found that even 'bug free' systems are full of bugs and must be tested. Testing must be strongly structured.

*Marc***.**
We have not seen the point of a lot of what we did until this year.

Found his client to be awkward: not returning calls/e-mails and not knowing what they want.

*Paul***.**
This is not a course on starting ones own business, but rather a course on how to work for someone else.

*Andy***.**
Learned the usefulness of classes and used them rather than just programming each part individually.

Anthony.
Learned that document templates are vital for keeping track of the large quantities of paper work generated in industrial projects.

Learned how to write and present reports.

Projects can and should be rejected by the students if they are unhappy with them and not simply accepted because the lecturer has suggested them.

Projects nearly always take longer than expected.

Suggests that the posts of Project Manager, Document Manager and System Administrator are needed and should be rotated between the students.

Brian.
Recommended a piece of commercial software to a client only to see it abandoned by the manufacturer. This was a useful lesson.

Testing should not be done by the writer, ‘Having someone else testing the software was a rude awakening and the number of bugs we found alarmed me given the simple nature of the program …’

It is important to find out who the actual client is.

Setting up the group network is very different to setting up one for yourself.

Define small, very specific groups with responsibility for individual non-overlapping goals.

5.3. Comments from clients.

Industrial advisory company specialising in consultancy in the field of metallurgical processing.

Client: Barry, Managing Director.

The students produced a self-paced web-based learning package for the company who were in the process of introducing new technology.

“This was highly successful, popular with staff and is still used 12 months later for new staff.”

Barry was very impressed with the students’ professionalism, their attitude to dealing with changes and problems caused by delays at the company end in the installation of the systems and said that the quality of their product was excellent.

Status: Completed.

Government department - Occupational Psychology Unit.

Client: Mark, Head of Section.

Project: Investigation into suitable specialist web search engines and the implementation of a highly specific browsing toll for use in the Unit.

“System still heavily used by staff in the Unit who find it very useful.”

Status: Completed.

A major health charity - Wetherby, Yorkshire and Humberside region.

Client: Sharon, Chief Field Officer.

Project: To provide a database for field officers to record all contacts and donations details and to generate reports, receipts etc. - currently all paper based. Needs to be available on laptops and the central database has to be updated every day by each officer. Significant replication and integrity issues. CRC HQ (London) want to expand the system out into all regions and to interface it to their central records. Major issues relating to vast quantities of data and interfacing Access to professional databases as well as differences of opinion between regional officers and HQ IT directors.

Involved two Genesys teams, both very professional and enthusiastic. "Very helpful and dealt with a lot of difficult and changing problems superbly. Very, very pleased."

Status: Completed.

A company specialising in designing hospital facilities.

Client: Ken, Research Director.

The students were asked to investigate constraints-based CAD design with VR simulation. This was a feasibility study involving the development of a prototype solution.

"Very impressed with the quality of work and the short time taken to achieve a great deal. Have got very close to what the company wanted. Could be very significant to the company's future business. Some problems caused by lack of liaison between members of the company.

Very good technical knowledge, very keen."

Status: Completed.

References.

[1] A. Stratton, M. Holcombe and P. Croll, "Improving the quality of software engineering courses through university based industrial projects." in *Projects in the Computing Curriculum*, (Eds. M. Holcombe, A. Stratton, S. Fincher, G. Griffiths), Springer, 1998, 47-69.
[2] S. Price, The Sheffield University maxi project - the industrial project manager's perspective." ibid, 184-195.