



# Enterprise skills in the computing curriculum

*Enterprise has been part of the computing curriculum at the University of Sheffield for many years. We describe the software house, Genesys Solutions, which is run by fourth-year MEng and advanced MSc students as part of their degrees and operates which as a real business.*

## **A lack of enterprise**

A recent headline in the *Times Higher Education Supplement* (18 October 2002) – ‘Industry attacks IT tuition’ – provides a graphic illustration of a message that has been consistently voiced for the past few years. The latest expression of concern follows the publication of a report, *E-skills Regional Gap*, produced by the e-skills UK organisation, a body that represents

employers and government organisations. The report states that only a fifth of computer science graduates in the UK work in the information and communications technology (ICT) industry. It also criticises universities for outdated curricula and for producing graduates who need substantial training before they can be productive.

In a list of desirable attributes looked for in graduate recruits technological know-how was bottom. More important were key skills such as ‘... discipline, business acumen, teamwork and writing which were as likely to be found in graduates from other disciplines.’

This has repeated the findings of five years ago. The Computing, Software and Services Association (CSSA), which represents the main players in the UK IT business, carried out extensive surveys of their members’ opinions,

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.

which identified that a third of graduate recruits had no qualifications in ITC. Then, as now, many employers also complained that graduates had no understanding of the world of business and this was regarded as a serious weakness in the educational process.

Very little seems to be changing and this should be a major concern to all those involved in the education of computing graduates. Most universities will try to address some of the issues: for example teamwork and presentation exercises are now more widely undertaken in courses. However, this is nowhere near enough.

At Sheffield we have been trying to address these concerns for over 12 years. Initially the emphasis was on developing skills that are associated with the successful construction of a software solution for a real commercial client. These can be categorised as:

- **individual personal skills** (presentation and communication, planning and reflection, researching etc.)
- **technical skills** to do with software development technology
- **teamwork skills** (working with people, listening, discussing and agreeing, monitoring, leading and delivering).

The students develop these skills in their second year as part of our Software Hut module. This activity takes place as a core part of our degree. Students spend one-third of their time in the second semester of Year 2 working with a real external client developing a solution for them. They are exposed to the complete development process from initial analysis of the client's business context, through an in-depth process of requirements elicitation and definition, to design, implementation and eventual delivery of a working system. In this module a client, typically, works with five student teams, all of which compete to build the best system. Having chosen the best system for their business, the client is then able to purchase this for a nominal fee. The process teaches

students a great deal about communicating with a client; about organising and managing a software development process, often in the context of the clients changing their requirements as they think more about their businesses and the possibilities that some new software may offer; as well as focusing on high quality production. It is clear to the students that the finished product must be of high quality, fit for its purpose and extremely reliable – all very tough challenges in software engineering. The extra motivation that comes from working with a real client rather than a lecturer pretending to be a client is enormous. Students are very eager to see the clients choose and use their system and this becomes a major focus for the project.

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 one's 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.

### Identifying the enterprise skills

A key challenge for educators is to try to see how enterprise skills can be developed in our senior students within the degree programme. So, what are these skills? There are many: skills relating to identifying markets, planning the development of products and services that will prove profitable, looking after clients and customers, managing quality and cost and many, many more business activities. 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, for example, an industrial placement but it is not until these are faced that the most fundamental of business issues

will be addressed.

The future employers of students with these enterprise skills will find amongst their recruits a much greater understanding of the reality of business 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 article looks at the organisation and success of this activity over the last five years.

### The Genesys experiment

Fourth-year MEng and advanced MSc students undertake a large exercise, amounting to one-third of their total studies, that of running a software house. The student-run company is called Genesys Solutions. Currently, it involves 47 MEng and MSc 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, not by a series of lectures but by actually doing it.

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 MEng degree), i.e. approximately 400 hours per student over the year.

Students involve themselves in the following business activities:

- researching market opportunities for software products
- carrying out IT audits on behalf of



**Figure 2** Clients come from all over the country and may be involved in any type of business or organisation.

local organisations and preparing appropriate IT strategies

- acting as software/computing consultants to local organisations
- developing software for clients<sup>1</sup>
- maintaining software for clients, possibly including maintenance of former projects
- delivering training courses.

The company has a weekly board meeting, reviews progress with its contracts and decides on future business activities. We, as their lecturers, attend in order to provide advice where necessary but leave the decisions to the board. This is vital: the students are given a great deal of responsibility for the company – it is their company not ours – and this has been an outstandingly successful approach. The students really relish having this responsibility and their attitude to the company is exemplary. Each project team will also have many other meetings both as a team and with their clients. These meetings as well as the entire projects are properly documented to professional standards.

The company has its own laboratory and network of computers operated independently of the department's laboratories, paid for from company earnings. A team of students manages this and oversees all the infrastructure requirements as well as determining

user policies and other managerial activities.

The students on the Genesys course have full responsibility for negotiating possible projects with their clients, including all terms, costs and contracts. The lecturers provide contacts where necessary and the students must follow these up and eventually have to decide whether to proceed with a project and under what terms and at what price.

Potential clients also make contact directly with the company. As far as possible, the students are 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.

The aim of the Genesys course is to give the students experience in modelling business process, developing requirements 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 Genesys course runs throughout the academic year and MSc students can continue to work in Genesys during the summer as their summer project, thus providing some continuity. This is useful in securing further work: the CSSA describes after



**Figure 3** The Genesys web site <http://www.genesys.shef.ac.uk>

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.

### Company organisation and infrastructure

During the course of the Genesys module the students keep company records and prepare company reports as well as developing project documents and other consultancy reports for clients. In the last two years the company has adopted a radical new software development methodology – extreme programming (XP). This aims to produce high-quality software in the context of projects with rapidly changing requirements. Over the years we have faced significant problems with clients wanting changes during the course of a project. The rather bureaucratic design-oriented process that we had used previously was a serious obstacle. XP relies on a number of well defined activities, including developing functional and unit tests before any code is written, using these tests to direct the software development, pair programming (two people per machine) and a number of other practices. Extreme programming is designed to be highly agile and quality-led.

The way that the students have organised themselves into teams has evolved during the development of Genesys. Currently there are three types of team: a marketing team, a research and development team and seven project teams. Each team comprises four to five students. The project teams each have a client portfolio and are involved in one development project at a time. The research and development team reviews and develops company infrastructure, processes and software. For example, we have developed some strong software testing tools to support the company's activities. There is also a company intranet which provides

support for all company processes, documentation templates and other resources. The R&D team also provides independent reviewers and testers for the other teams' projects. This has increased quality considerably. Every deliverable item, which includes all documentation and all software, has to be independently checked. This applies to all versions of all deliverable items.

The rule of thumb used is that testing in general will take as much effort as development. Two students in R&D also take the role of systems administrators, having full responsibility for the support of the company infrastructure. These students produce monthly records of the system and archive and back up everything. The system is continually being developed, and company earnings are invested in the computing infrastructure.

The marketing team looks at past and current projects to identify opportunities for exploiting the software archive in new areas. We are trying to create an object repository so that new solutions can be developed and configured easily and quickly for new

clients.

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

At the end of the course a final report is produced, consisting of the project and company 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.

All this information is used in the assessment of the students.



Figure 4 A student pair programming

## Experiences from the Genesys project

Although the Genesys project has run for only five cohorts of students, there are already many conclusions that can be drawn from the experience. It is certainly nothing like anything any of us has ever done before, either staff or student. This, naturally, brings with it some problems. The company has evolved over this time as we learned 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.

### Developing a professional attitude

To their surprise the students learned that a professional attitude and image are among the most important assets a business can have. 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, careful planning that they had at first thought was just a waste of valuable working time. Addressing issues such as coding standards, effective testing and review has also had a big impact.

Another key benefit is the development of a customer-oriented culture. Customer/client care is vital to any successful business and the students are in the front line as far as this goes. This is a unique experience.

The future employers of Genesys students will be taking on fresh, keen young graduates but will not have to spend so much time and effort acclimatising them to professional attitudes and approaches because these 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 many employers mention as being lacking in graduates. This understanding may help the students to appreciate the decisions and attitudes of their superiors, thus reducing friction and increasing the scope for cooperation 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 successes and failures. The groups recognise where 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. Since we claim that Genesys is as close to a real software house as we can practically 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.

### The business imperative

The emphasis of much of the engineering curriculum is on the professional processes of design and construction – emphasising quality processes, systematic analysis, professional documentation, etc. The business perspective, if presented at all, is often somewhat remote from the reality of the practising engineer. Case studies are a common way to address some of the business issues in class but cannot compete with a real problem that needs to be faced by the class. In Genesys we meet, directly, real business dilemmas. One of our current projects involves the development of an innovative software product for a client. The software needs to be demonstrated to the client's potential

business backers within a short period of time. Initially the students started with an in-depth requirements definition and analysis stage because their training had emphasised the need to engineer things properly, to provide a firm basis for quality assurance and the maintenance of the software. However the client was concerned about this approach. The real need was to build a quick and dirty demonstration package rapidly. If the demonstration was successful there would then be money to develop the software properly; if not, there would be no contract and thus no project. This experience taught the students a little about the realities of business life. Ultimately, if a product does not make money then it is not going to be successful. There could be many reasons for this. For example, even a great product, if it is marketed at the wrong time, can fail. Markets can be seized by an inferior product which delivers enough to be successful, and then generates income that can be invested in product improvement without losing market share.

### Genesys: the benefits for staff and for clients

The staff supervising the course also gain from:

- 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
- exploring the strengths and weaknesses of current design methodologies.

All of these have 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.

In terms of my own particular research it has provided me with the opportunity to try out some of our state-of-the-art software testing

techniques in realistic settings. We have also incorporated Genesys into our newly formed software observatory, alongside the second year Software Hut so that we can observe how different design methodologies work out in practice, in some cases in comparative empirical studies. There is a tremendous need, in a hype-driven subject, to obtain real scientific data about the effectiveness of different software development technologies. These need to be tested in laboratory conditions but also under a realistic market environment. Since requirements change is a major problem in IT developments it is vital that clients are involved in these experiments. The observatory allows us to do this, to evaluate different approaches and to see how new technologies can be adopted in a working company.

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 a low-risk prototype or consultancy. They also learn a lot about their company through interacting with bright, questioning professionally trained students as well as learning the role of being an effective client. This is an important skill which is vital for a successful project.

## Educational evaluation

The educational value of the enterprise is potentially difficult to assess since the main benefits will probably come to light years after students have left. However it has been possible to see that students have obtained a much more mature understanding of the business environment, of working with real clients 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 one's 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 use of extreme programming (XP) as a design approach has produced both benefits and problems. It is by no means easy to implement all the practices fully and current research is focusing on what the key aspects are and how the difficult activities can be made easier.

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 and much more important than the techniques, tools and notations used.

## Conclusions

Enterprise can be taught successfully in universities but the way it has to be done is radically different from our normal educational processes. Educationalists have to be bold and adopt a similar real-life oriented strategy. It takes courage but the learning experience for academics is amazing.

However, there are responsibilities for employers here, also. Sometimes our enterprising graduates join companies with great expectations, only to be disappointed when the companies do not allow them to express their skills and they become

bogged down in bureaucratic cultures. In such cases, and they have happened, the students soon leave and join smaller, more dynamic businesses. There is thus a two-way responsibility: we can produce more enterprising graduates but some companies need to change to get the best out of them.

## References

- 1 Stratton, A. Holcombe, M. and Croll, P. 1998, 'Improving the quality of software engineering courses through university based industrial projects' in *Projects in the Computing Curriculum* (Eds. Holcombe, M. Stratton, A. Fincher, S. Griffiths, G.), Springer, 47–69.
- 2 Beck, K. 1999, *Extreme Programming Explained: Embrace Change*, Addison Wesley.

---

**Mike Holcombe CEng, CMath is Professor of Computer Science and formerly Dean of the Faculty of Engineering at the University of Sheffield. He is a Fellow of the British Computer Society and a Fellow of the Institute of Mathematics and its Applications. He has research and teaching interests in software engineering particularly testing, theoretical computer science and computational biology.**




---

**Marian Gheorghe holds both a BSc and a PhD in computer science and mathematics from Bucharest University. Over the last ten years he cooperated with a number of universities in the UK, Germany and Greece in teaching and research areas. Dr Gheorghe is now a lecturer in the Department of Computer Science, Sheffield University where he teaches methodologies for software analysis and design.**

