

# Learning the Business

Mike Holcombe, Marian Gheorghe

Department of Computer Science, University of Sheffield, Sheffield, UK  
Email: [m.holcombe@dcs.shef.ac.uk](mailto:m.holcombe@dcs.shef.ac.uk)

Received November 7<sup>th</sup>, 2011; revised December 8<sup>th</sup>, 2011; accepted December 28<sup>th</sup>, 2011

Developing software is a highly creative process. This paper describes a novel approach to teaching software engineering which involves university students working in partnership with external clients from business, charities and the public sector building solutions to their business and other problems. The paper describes the basic principles behind these activities and focuses on the experiences of teaching advanced students through the medium of a commercial software development company specifically set up to be run by the students as part of their degree course. The evidence from student and employer feedback demonstrates that this approach has been highly successful for the past 20 years or so but despite this it has rarely, if ever, been replicated elsewhere.

**Keywords:** Software Engineering; Commercial Clients; Projects; Enterprise

## Introduction

Genesys Solutions is a software house with a successful list of commercial clients developed over the past 16 years. It is entirely run by senior students, 4<sup>th</sup> year undergraduates and MSc students, in the Department of Computer Science as part of their computer science degree. Students are responsible for marketing and sales, all software development and the management of the company's computer infrastructure. This experience enables them to undertake the complete range of activities that any successful software company carries out from consultancy services; contract negotiation; development of high quality solutions for their clients; customer relations; software maintenance and the setting of company strategy and business planning. All of these activities require a challenging mix of creative, logical, quality-oriented and entrepreneurial skills.

Designing software to solve real problems is a difficult creative activity. In general, software engineering follows an approach similar to the Creative Problem Solving Process (CPS), [Osborne & Parnes (1950)]. CPS is a structured method for generating novel and useful solutions to problems. The first phase is *Exploring the challenge*, then the *Generate ideas* stage follows and finally the *implementation and evaluation of the solution*—in this case a software application. This has to be done in the highly constrained and often unforgiving environment of a programming language. This software engineering involves both creativity and discipline with quality assurance a vital underpinning theme.

However, the traditional CPS approach doesn't work in situations where the problem is ill-defined. This is usually the case for most software development projects. The process must be refined and made much more iterative, culminating in the agile process that we now use. Initially the developers—students in this case—talk to the client about their business or organization and their objectives. This discussion then starts to focus on what possible solutions might look like. This often includes some rapid prototypes and screen shots to enable both sides to understand then issues better. A set of tests are prepared to enable both sides to see if what is being done works.

This emphasis on how detailed evaluation will be done at the start of the project is very important. Then a development stage could be carried out and further reviews and exploration carried out, thus repeating the process. This might lead to significant changes to the project and is a key aspect of agile development. The client's business environment might change and this could lead to changed requirements and the decision to throw away some of what has been developed. Such an approach brings with it very strong demands and challenges for students but is a good experience in terms of demonstrating that the real world is messy, uncertain and volatile—something that they will have to deal with in later life.

This emphasizes that creativity, in reality, often exists in a dynamic environment and adaptability and quality assurance are key issues to be mastered.

For many years the Department of Computer Science at the University of Sheffield has put great emphasis on the development of team skills in realistic settings—in particular the second year Software Hut module involves teams of students working with external business clients developing a software solution for their business or organization [Kalra et al. (2005)]. This is carefully managed with the aim of developing a professional attitude in the students and using a management framework and quality development tools that usually ensures success as well as a powerful learning experience for the students. This has been running since 1987 and was highly praised by the UK Government's Teaching Quality Assessment visit in 1994. The overall undergraduate programme places group projects at the core of the curriculum. Thus students, on their first day at the University, are placed into teams and spend one sixth of their first year going through the main stages of the software engineering lifecycle. Thus they learn about business analysis and requirements capture, write a short "White Paper", develop a requirements document and an acceptance test report for a "pretend" application over the first few weeks. The teams then exchange these documents and receive the documentation about a different application from another team ready for the next stage, which is the development of a design document. This

takes place in Semester 2 and is followed by the implementation stage—again swapping applications to a 3<sup>rd</sup> example. Finally they receive the documents and code for their original system and evaluate it, writing a report on this stage. This experience is intended to underline the importance of coherent, informative and concise documentation and provides them with some wider-ranging experiences of building a piece of software from concept to delivery. This is the foundation for the 2<sup>nd</sup> year Software Hut experience where they are working with a real external business client.

In year 3 the students carry out an individual research project alongside their advanced technical courses.

Following this, a grant was awarded (under the Fund for the Development of Teaching and Learning) to develop the ideas further. This coincided with the UK Government introducing the SARTOR (Standards and Routes TO Registration) for the engineering profession that required the introduction of a 4<sup>th</sup> year of study for engineering accreditation. It was decided to introduce a radical new approach into the 4<sup>th</sup> year that built on the Software Hut experience.

A company was set up within the University which would carry out commercial software development activities and which would be largely run by the students as part of their 4<sup>th</sup> year studies. This article is a record of some of the activities, successes and lessons learned during the past 16 years.

### **The Early Days of the Company**

Initially, the numbers of students taking the 4<sup>th</sup> year were a minority – less than a dozen, so the company started small and this provided the academics with an opportunity to learn quickly and without too much embarrassment! The first issue was to find some premises in which to house the company and some computing infrastructure. This was achieved by the use of a small laboratory in the Department and money from the FDTL grant. At the beginning these machines were managed by the Department's technical support and the company concentrated on getting business and developing both a set of company processes and a good reputation. The students were given a great deal of responsibility for these issues and this worked out very well as they were both motivated and sensible about what was introduced. The students negotiated contracts and set the price of projects with advice from us. The general policy was to charge at a rate that the clients could afford rather than at commercial rates. This allowed us to develop a niche market that did not compete with established software companies. Genesys has provided many small start-up companies, charities and others with an opportunity to have bespoke software created for their businesses and organizations that they would otherwise never have been able to afford. This in no respect made the intellectual challenge of the projects different from a full commercial one.

In fact, the decision to give the students a lot of responsibility for running the business has proved to be a very successful one. Initially the main focus of activity was on producing fairly standard software systems such as databases and planning tools for a variety of organizations—small business and charities and public sector departments with the health service providing many such projects. These projects used a fairly standard software development approach with the students working closely with their clients and managing the projects. As their supervisors the academic staff met the teams regularly and provided

support, training and advice. The company had regular board meetings that were chaired by a student and the emphasis of the meetings were to provide a means of sharing information about the projects and the technologies used, deciding on which future contracts to accept and planning out the company strategy. This latter included: defining the company profile, corporate image, web site as well as the internal processes, quality assurance, standard tools and templates. The students had many good ideas and usually made best use of the latest technology.

Business was fairly easy to come by—it was mainly through word of mouth or contacts inside the University and research collaborators in other universities. The new course was becoming more popular and reached a point when there were 50 students in the company. They spent one sixth of their time doing the module and it soon became clear that this was not enough—both the students wanted to spend more time in the company and the business was there to justify it. One other factor was that there was some risk that the enthusiasm for the company meant that some students spent more time on this work with the result that their other modules suffered. We thus agreed to double the time for the course—this meant that the students were expected to spend 15 hours per week—one third of their study time. This has remained in place and has been a positive development.

By this time we had our own premises away from the Department and our network was completely independent from the University's. This meant that some of the students had to become fully-fledged systems administrators running a network of up to 30 PCs with full file store, software infrastructure, internet connection, a highly secure system and full 24/7 back-up. I have always been impressed at the professionalism and technical skills that these students display—in many cases the systems they run are better—more modern, more reliable and generally provide a better service than some offered by full-time professional services!

At the end of their studies the company board meeting, including the students, looked at the balance sheet and the students decided what to do with profits—typically this involved holding back some for reinvesting in the company infrastructure with the rest being split amongst the students as a cash payment. Of course, the students were also getting academic credits towards their degrees.

We have also had a long-term relationship with IBM (Hursley Park) who have provided three experienced developers as mentors for many years. These mentors come up every term and provide advice, training courses, as well as “adopting” some of the teams having telephone conference calls and e-mail discussions at other times. This is highly appreciated and the benefits are in both directions.

### **Becoming a Fully Constituted Company—The Next Stage in the Company's History**

The students were running the company during term time but in vacations there was a gap. On occasions there were opportunities for business over the vacations that we couldn't take except at the cost of employing students after they had graduated—this we did a few times but the administrative load was rather heavy. There was also the issue of maintenance which did arise, usually it was for an extension to a client's system but sometimes it was necessary to fix some problem. With an “empty” company over summer this was a potential issue.

The decision was taken by the University to spin out the company with a slightly different role. Some funds were available from the European Union to support this process. The new company—called *epiGenesys*—was duly constituted and registered with the Government. A legally constituted board was established with the University being the sole shareholder.

A major role for the new company was to support the University’s exploitation of its research Intellectual Property through the transformation of experimental software developed by research projects into commercial quality products. In order to achieve this some permanent employees were appointed from amongst the best of the Department’s students and graduates. Currently there are eight staff. They have a dual role—managing the *epiGenesys* project list and mentoring the students in the *Genesys* company—*Genesys* is regarded as a sub-department of *epiGenesys*. To support the permanent staff the company faces a challenging business revenue target which has concentrated all of our minds.

### Early Adoption of Agile Development

It became clear around the turn of the century that the “traditional” design-led approach we were taking in the company was not as appropriate as it might have been. Many of our clients were unsure of their precise system requirements and the process of business analysis and requirements capture was a rather dynamic one. The emergence of Extreme Programming (XP) and the work by Kent Beck [Beck (1999)] came at just the right time—we adopted XP in 2000. This has been a great success and we have increased our efficiency and quality substantially through this. We undertook a large funded research project into the comparison of the agile development approach compared with the traditional software development one using the extensive project data we collected, including a large number of comparative experiments where different student teams used the different methods on the same projects. This produced a clear advantage in terms of quality of delivery for the agile

approach. It is also greatly enjoyed and preferred by students and this was established through measuring well being during the projects as well as other samples of student attitudes.

As we explored the detailed processes of XP we developed our own software infrastructure and management approach. This is the subject of the recent textbook [Holcombe (2008)] which tries to look at software engineering in a broader than usual perspective with an emphasis on “people” issues.

### The *epiGenesys* Way

The basic software development approach of the company involves a rigorous version of eXtreme Programming (XP). As far as possible students work in pairs on the projects. The pairs change around regularly. Although many students were sceptical about this initially most recognize the great benefits that it brings once they get involved in a challenging project. The initial phase involves meetings with the clients and the analysis of their business needs and context. This is done in a number of ways—interviews, questionnaires, role playing etc. and the outcome within a week or two is an outline requirements description and a set of stories. These represent small elements of functionality that can be implemented in a week (15 hours of pair working). A tool, *StoriPost*, has been developed to provide support for the management of the stories. A screenshot of this tool is in **Figure 1**.

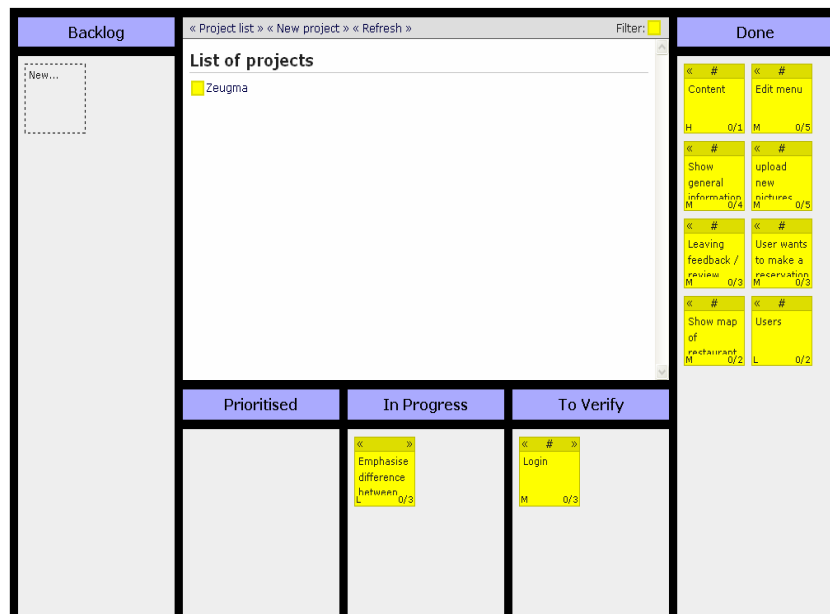
Each story is described according to the template:

As each story is prioritised, worked on, tested and completed the story “postit” is moved right to the next column. The details of each story can be opened up as in **Figure 2**.

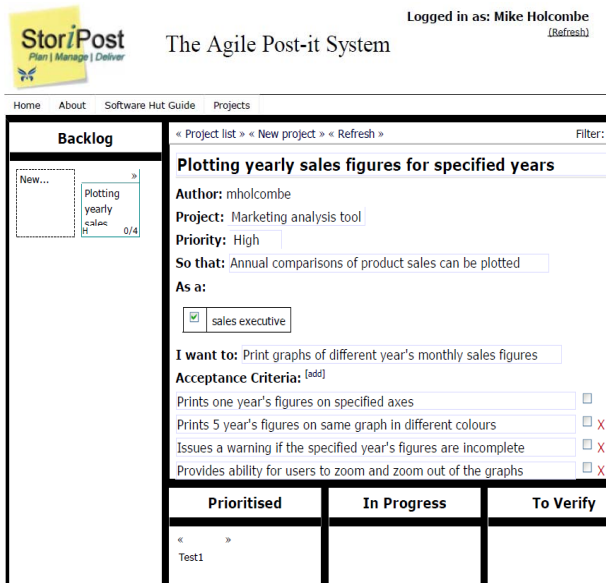
It is important for a company to develop its own tools and processes to both improve productivity but also to build a company culture. *StoriPost* is one example of this.

A dynamic summary of the stories and solution is generated as a lightweight requirements document since many clients require such a statement.

Depending on the type of application frameworks such as



**Figure 1.**  
StoriPost output.



**Figure 2.**  
StoriPost templates.

Ruby on Rails, and in-house systems developed in epiGenesys such as PHP-trax are used. Use of Subversion (svn) a version control system is mandated. Continuous integration is supported using cruise control.

Testing is taken very seriously and we try to use test-first techniques, which means writing the tests before coding. The acceptance criteria specified in the stories determines the unit tests. Testing tools such as phpUnit are used for unit testing the stories. An architectural design notation—XXM (extreme X-machines, Holcombe (2008)) is a popular way to illustrate how the stories are integrated through a user interface. An example is shown in **Figure 3**.

Systems test—particularly those systems with a web front end—use Celerity and in-house feature converters using story and XXM information to automate the filling of web forms and actions in testing.

Other aspects of the software development are covered by appropriate tools: processes are handled by Basecamp, other types of testing by Cucumber, training Genesys students at the beginning of their programme with in-house built tools.

## Typical Projects

A brief selection of projects includes:

- 1) Bummitt—a social networking web site where teams of students competing in a charity dash to see how far they can get in continental Europe on £15 and which shows maps of progress based on receiving text messages from the teams;
- 2) Iceberg tracking software for shipping based on some prototype software developed by a research team;
- 3) A web site for children suffering from Cystic Fibrosis that helps them, through the use of a game, to calculate their medication based on what they have eaten that day;
- 4) A video game about the 100 years war in France in the middle ages that features manuscripts, weapons and other artifacts being showcased in a museum exhibition;
- 5) A system that allows the audience at a meeting to vote on questions posed by the presenter using the Bluetooth capabili-

ties of their mobile phones.

There are many other examples.

## Lessons Learnt

It is quite clear that this activity greatly increases all the students' motivation and develops their skills in many areas to an extremely high level. They are real professionals when they finish and can walk straight into an industrial position and be productive immediately. Very few graduates are in this position. All the graduates from this course are highly sought after by companies large and small.

The popularity of the course is extremely high with well over 90% of students confirming that this was the best thing in their degree programme and stating how much they enjoyed the experience even though it was very challenging.

All our evidence indicates that the course meets the needs of both students and employers. Not only that but some of the students go on into academic research and make that a great success as well—they are highly skilled, highly motivated and very well organized—key attributes for a successful researcher as well.

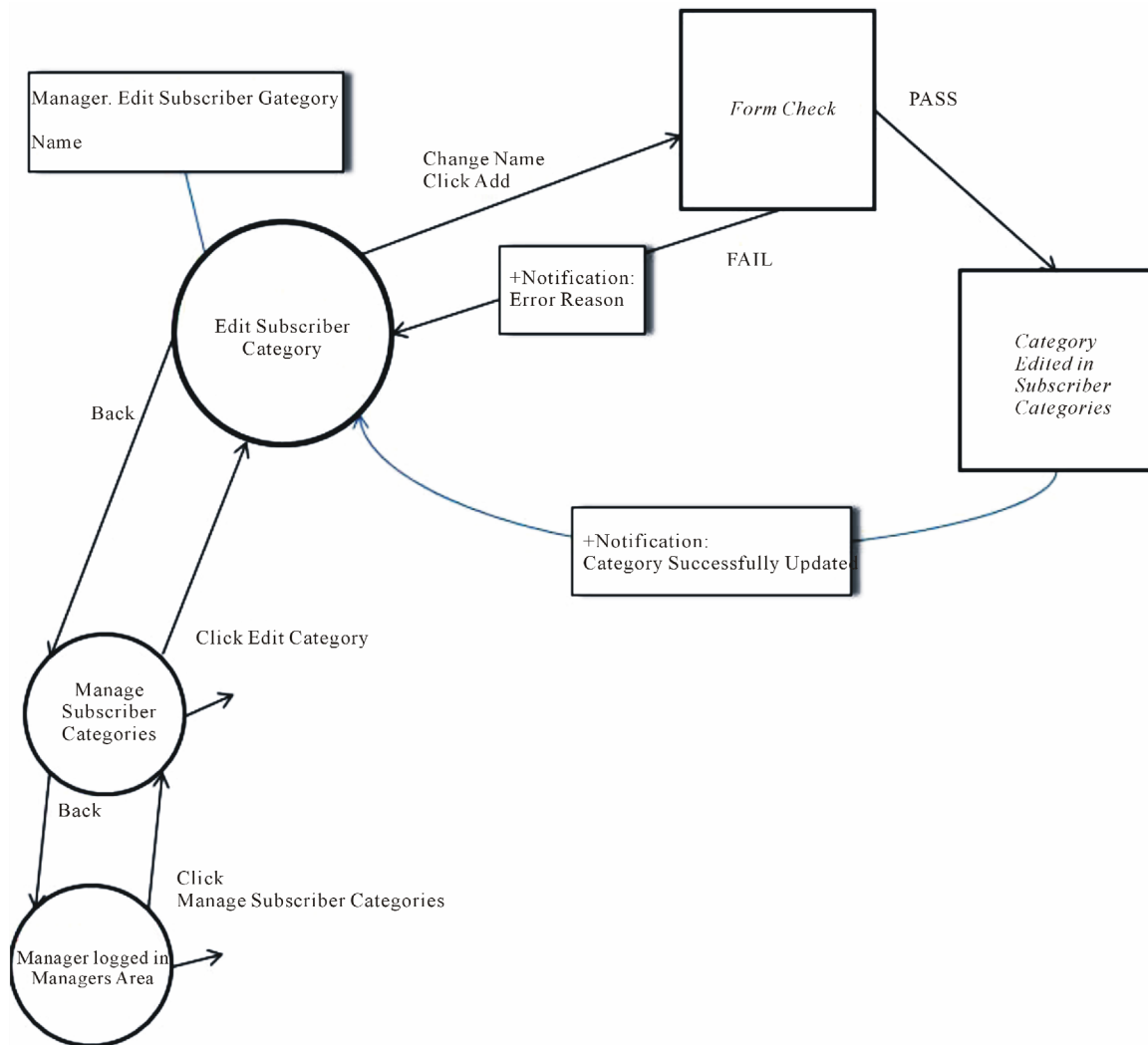
## The Benefits Are for Everyone—Win, Win, Win, Win!

The benefits for students are clear from the above, but there are also benefits for the academics involved in that it forces us to address the realities of real software development in a commercial context and demonstrates that much academic thinking in software development is based on myths. Our research has benefited because we can carry out empirical research into development methodologies, study team related issues and base this work on real commercial development projects. Much of this experience has been incorporated in [Beck (1999)].

Some universities encourage or mandate students to spend some time on an industrial placement. These can be very rewarding experiences during which the students can gain many new skills. However, such placements can vary greatly in the quality of the experience for the student. In some cases they provide an excellent vehicle for improving their knowledge and professionalism. However, in some cases the students do not gain as much as they could. I have heard of many cases where the students have been given mundane tasks and little opportunity to engage with an important project. They rarely meet with clients, for example. They also rarely have any strategic input into company policy or the opportunity to take major responsibility for delivering a successful product. All of these aspects form part of the Genesys experience. One of the members of the Department's Industrial Liaison Board once commented that there were two types of software engineers working in industry—the foot soldiers (programmers) and the leaders—project leaders, business leaders etc. One of our targets is to educate for the latter posts as well as the former.

The local community also benefits, there are many small start-up companies that have been able to progress using software developed by the students—this includes charities and public sector organizations as well.

The University benefits because it is seen to be doing something useful for the community and it is attracting a lot of good students—unlike many other universities Computer Science at Sheffield is booming in terms of student recruitment.



**Figure 3.**  
An XXM from a project—see (Holcombe, 2008).

The United Kingdom Government collects various performance statistics about university courses and these inevitably find their way into league tables published by the media. One aspect is graduate employment and this is measured by the Graduate Destination survey—all graduates are contacted 6 months after graduation and recorded as 1) being in graduate level employment; 2) undertaking further study or 3) not in either of these. Our students have consistently been reported with an unemployment rate of close to 0% for the past few years placing the Department at the top of the list.

Another major survey is the National Student Survey. Here students are contacted on behalf of the Government by a survey firm and asked about their satisfaction with various aspects of their degree course. Again, every course in every university in the United Kingdom is surveyed and in the 2009 survey the University of Sheffield was top for overall course satisfaction in Computer Science gaining a 100% rating. There are 140 institutions offering computer science and related degrees in the United Kingdom and none others received this rating.

There is thus strong evidence that our approach to practical project work is a major contributor to these results.

## Conclusion

A question we are often asked by business executives, politicians etc. is—why doesn't everyone else do something similar? There are a number of reasons for this—it is difficult to start up something as innovative and unprecedented as this because of the risk aversion and bureaucratic constraints that university teaching experiences currently. Few academics have the experience to try something like this on, it seems. All of these obstacles applied to us when we started but, perhaps through naivety or just sheer bloody-mindedness we just got on with it and made it work. There were lots of problems on the way and we learnt rapidly but it was certainly worth it in the end.

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 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 the bureaucratic culture of the company. In such cases, and they have happened, the students soon leave and join smaller, more dynamic business. 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

- Osborne, A., & Parnes S. J. (1950). *Creative problem solving: Resources for CPS practitioners*. Charlotte, NC: OmniSkills, LLC.
- Beck, K. (1999). *Extreme programming explained: Embrace change*. New York: Addison-Wesley.
- Holcombe, M. (2008). *Running an agile software development project*. New York: Wiley. [doi:10.1002/9780470385883](https://doi.org/10.1002/9780470385883)
- [http://www.engc.org.uk/UKSPEC/SARTOR/sartor\\_executive\\_summary.aspx](http://www.engc.org.uk/UKSPEC/SARTOR/sartor_executive_summary.aspx)
- <http://www.heacademy.ac.uk/ourwork/networks/fddl>
- <http://cruisecontrol.sourceforge.net/>
- <http://www.phpunit.de/>
- <http://www.dcs.shef.ac.uk/~cthompson/xxm/>
- <http://agile.genesys.shef.ac.uk/wp-content/uploads/2008/07/hut08sxxm-wiki.pdf>
- <http://celerity.rubyforge.org/>
- Kalra, B., Thomson, C., & Holcombe M., (2005) The Software Hut—A student experience of eXtreme Programming with real commercial clients. *Extreme Programming and Agile Processes in Software Engineering Proceedings*, 3556, 323-324.