

# Internationalising the Postgraduate Curriculum: a Ten year Case Study

Dharmendra Shadija<sup>1</sup> and Richard Hill<sup>2</sup>

<sup>1</sup>Sheffield Hallam University, Sheffield, UK

[d.shadija@shu.ac.uk](mailto:d.shadija@shu.ac.uk)

<sup>2</sup>University of Huddersfield, Huddersfield, UK

[r.hill@hud.ac.uk](mailto:r.hill@hud.ac.uk)

**Abstract** We describe the experience of creating and developing a postgraduate computing provision, in response to a set of different learning needs from large cohorts of students recruited from the Indian Sub-continent. Amongst a number of changes to the curriculum and delivery, two particular aspects of the teaching stand-out in terms of their contribution to enhancement. First, a shift in emphasis to teach software design over implementation. Second, a delivery ethos was modified to encourage professional development through the facilitation of student-led curricula, leading to a considerable improvement in student participation. Over the course of a ten-year period, student attainment and engagement increased significantly.

## Introduction

The development of the internet and associated technologies has created a demand for highly-skilled computing professionals to be able to design and implement software applications that are resilient, responsive, interactive and can reliably exploit the scalability requirements of distributed computing infrastructure. Software Engineering is a mature domain, and the methods and approaches that have evolved through its study have addressed many of the challenges of creating large-scale software applications, especially amongst teams

of software developers, with all of the associated challenges that group working can present.

Nonetheless, there are three forces evident that share a similar outcome. First, the computing industry wants to recruit software developers that require minimal training in the practice of software engineering. Second, aspiring software developers want to minimise the time spent studying in order to gain employed experience.

Finally, the pervasive adoption of distributed computing and all of the opportunities that it offers, means that technology companies want to employ computing professionals who can understand the design and application of web-oriented software engineering. These professionals will use the knowledge to advocate, consult and sell technology, with no obvious emphasis upon the acquisition of programming skills.

All of these scenarios mean that there is a significant demand for optimised educational paths that offer the fastest possible route to employment in a field of computing.

Alongside this, there was an emerging market for UK Universities, for international students, who already held undergraduate degrees, to seek out postgraduate programmes in order that they could develop deeper knowledge. In 2008, the inclusion of a two-year post-study work visa meant that the UK was an attractive proposition for international students as they could study an award at level 7 and then acquire work experience over a concentrated period of time, before either returning to their home country, or securing further employment in the UK or elsewhere.

This chapter focuses on the development of a specific module that was designed as part of a postgraduate computing programme. The influx of large, recurrent cohorts of international students from the Indian Sub-continent presented new challenges for the module, that are explained and discussed as a result of ten years of delivery.

## **Web application curriculum**

Web Application Design and Modelling (WADM) was originally developed for a technical postgraduate degree with the title “MSc Multimedia Technologies for E-Commerce”. The module was conceived in 2003 and was one of a collection of modules that were primarily focused on the teaching of advanced technical skills for either organisations who desired “back-room” staff, or for entrepreneurial individuals who expressed a need to learn specific expertise in order to support their business aspirations. The aim of the module was to deepen the knowledge and insight into the whole life cycle of an internet development project using Object Oriented techniques.

The module had been delivered for a number of years to small cohorts of students with diverse demographics, and through the annual process of module evaluation and review, had undergone a series of enhancements that resulted in students’ performance on the module being consistent with their attainment on other modules.

One particular area of the module that appeared to be more challenging was the students’ acquisition of design knowledge. They consistently struggled with managing the uncertainty of requirements specifications, and there was a tendency to ‘hack’ together a solution, that whilst might have been functional, was not sufficiently robust enough to be scalable. Students always required more support to be able to build their design skills and although they met the learning outcomes, it was rare to see any excellent examples of design within the submissions.

A portfolio review that included an evaluation of emerging international markets, resulted in the introduction of a suite of postgraduate ‘professional’ courses. These courses were designed to offer an increased appeal to overseas applicants, and used a modular structure to combine efficiencies in delivery, with specialist pathways - IT Professional; Database Professional; Computer Networking Professional - so that students could align with a route that matched their interests.

WADM was selected as a core module that all of the learners would experience as part of their respective MSc programme.

In anticipation of the new international cohort, which differed both in demographic and volume, the teaching team concentrated upon planning for the changes in delivery such as the use of a large lecture theatre, managing many tutorial groups, and providing transitional support for the added challenge of learners studying within an educational culture that was new to them.

### **Module design**

The academic aim of the module is to develop knowledge and provide insight into the whole life cycle of a computer system development project using Object Oriented (OO) techniques.

By the end of the module, learners are expected to achieve the following learning outcomes:

- Formulate a 'whole system' view;
- Appraise and model user requirements;
- Explain Object Oriented principles (such as encapsulation, inheritance and polymorphism) and apply software reuse techniques (including the use of libraries);
- Identify and explain the main aspects of suitable design notations such as UML and use them to model systems;
- Implement a software system using the Object Oriented features of a suitable programming language such as Java;

Initially, the intention was to simulate an experience of delivering a software project in industry. Students were expected to design their applications using OO concepts, and then implement a working prototype. The module started with the delivery of OO concepts via lectures and tutorials, based on pre-written requirements. During the very first tutorial session, learners were required to identify use cases and producing sequence diagrams based on their use case models.

The second session (week three) was focused on the production of class diagrams, and was supported by lecture materials describing

OO concepts and the discovery of classes from requirements documentation.

After the initial sessions on design, the focus then shifted towards web development, starting with HTML pages and then progressing to using Java to automate the creation of HTML.

Java teaching commenced with the writing of console-based applications ("Hello World" programs), then input and output on console using core Java constructs, and finally the collection framework. The remaining three weeks of the module were focused on developing web applications using JSP, Servlets and JDBC.

Week	Lecture	Tutorial	Assessment
1	Module Introduction	Introduction to case study	
2	Use case diagram and Sequence diagrams	Designing Use case and Sequence diagrams	
3	OO Concepts	Designing Class diagrams	
4	Introduction to HTML	Designing Web pages using HTML	
5	Introduction to Java programming	Writing "Hello World" program in Java	
6	Input and Output in Java	Write a program which accepts input from console and displays output on console	
7	Introduction to collections framework	Writing Java programs using collection framework classes	
8	Developing JSP pages	Developing JSP pages	
9	Building 2-tier apps with Javabeans and JSPs	Building 2-tier apps with Javabeans and JSPs	
10	Connecting JSP pages to database using JDBC	Connecting JSP pages to database using JDBC	
11			Final team assignment submission and online test

**Table 1. Initial schedule for Web Application Design and Modelling module.**

## **Motivations for change**

As expected, the first cohort demonstrated a marked increase in student numbers. This was compounded by a shift from one intake per year, to two, which made further demands upon the teaching team to rationalise and optimise delivery to suit the increasing scale of operations.

Several intakes during the first year enabled a more comprehensive evaluation to be conducted across two cohorts. The compression of multiple instances of delivery permitted a greater deal of flexibility to experiment with different aspects of the delivery, as feedback was more prompt.

After two years of delivery, a considerable number of minor changes had occurred that were attributed mostly to dealing with the large volume of learners. What persisted were the following themes:

- The quality of the design work that was produced was at best basic, and this re-emphasised the challenge that had been experienced on the module in prior years prior to the increased cohort sizes. Since the recruitment was at scale, so were any problems, and it became an area of significant concern as it was not possible to provide the same amount of support for the sheer numbers of learners with difficulties;
- Plagiarism on team assignments was prevalent. From the perspective of university regulations, the plagiarism was categorised as collusion, which incurs the most severe sanctions. From a staff perspective this was somewhat demoralising, and as additional staff were added to the original teaching team to cope with increased recruitment, there was an emerging perspective that the cohorts had a predisposition to cheat;
- Students were finding it challenging to interpret the requirements of the assignment case study. As part of the coursework, students were provided with a pre-written case study, and the tutors played the role of the 'client' who wanted the eventual web application. Case studies were contemporary and deliberately kept current with popular web applications so that all students would have

some experience of using such an application. However, the students brought their own experience to bear in such a way that they imposed the requirements of web applications that they had used onto their submissions, rather than using their experience to interpret the case study that had been provided for them. As a result, some students did not fully account for the requirements that were expressed in the assignment case study. A professional software engineer or business analyst would ask a range of questions of the client to ensure that they had understood the requirements properly, and the validation of the client's needs and wants was not occurring;

- Student engagement was noticeably poorer with the international cohorts. More students found it challenging to arrive to taught sessions on time, often missing valuable instructions that were delivered in the first fifteen minutes of class. Additionally, the majority of students were also undertaking employed work outside of the university, involving significant quantities of hours that were often un-social, which impacted upon their ability to study independently or meet with team members outside of formal scheduled classes. Students would usually turn up for first couple of sessions but the attendance at lectures would usually drop to less than 50%.

One of the aims of the module was for learners to be able to produce design documentation such as UML use case, class and sequence diagrams. We observed that consistently, the student's design documentation lacked detail and coherence.

A perennial challenge with technical computing curriculum design is balancing the technical content delivery against the time available to practice application of the knowledge. There is the additional tension between what students expect, versus what industry wants and academia provides.

Employers consistently ask for students who have the skills to be able to design software, and they are less concerned with particular programming languages have been learned. Industrial employers ar-

gue that once the core skills have been established, a change in programming syntax can be taught on the job quickly. Academia supports this by focusing on the teaching of design, documentation and refactoring skills, which can apply to any programming syntax.

However, smaller employers may have a specific need for an employee with specific experience in a programming language, and coupled with students who want to maximise “value for money” by studying as much as possible on their programme, both sets of stakeholders often want to see lots of content.

This is a perennial challenge for academia and it is a difficult balancing act to satisfy multiple, conflicting objectives. In this particular case the students that were being recruited did present a more focused ambition.

Many of the international students had a common motivation for overseas postgraduate study. The attainment of an award from the UK enhanced their opportunities for highly-skilled employment in their home countries. In addition, there was a perception that their profile was enhanced further if they successfully gained employment within the UK, prior to returning home.

At the time, there were schemes such as ‘post study work visas’ that facilitated overseas students to solicit UK work experience that was related to their course of study.

### **Back to basics**

University annual monitoring and review processes inevitably reported the issues that the teaching team had already identified (and described in the previous section). The teaching team felt that there was a risk that if the individual items of concern were dealt with, as might normally be the case as a result of a module review, the desired improvements would not materialise.



Using an Appreciative Inquiry approach (Cooperrider and Srivastva, 1987), a series of discussions were held, first amongst academic staff, and then subsequently by drawing in contribution from students, staff involved in international recruitment, student support staff and also course alumni. What emerged was a wealth of insight into the issues that affect international students who have chosen to study in the UK.

One aspect that was significant was the almost universal need for such students to be employed during their studies so that they could afford to live. From a pedagogic standpoint there can be a difficult tension between the need to devote time to full-time study, as opposed to time spent in employment during the course of study itself. Many of the students were obtaining work that occupied anti-social hours, which created an additional strain for those that were scheduled to attend lectures and classes early the next morning, after a late night shift.

Pressures such as these inevitably draw out strategic behaviours in learners, who are having to balance competing priorities. Such behaviours can then manifest themselves in groups of students cooperating to complete assignments, which runs the risk of accusations of collusion.

An additional effect of social collaboration was the temptation for groups to organize for representatives to attend classes and retrieve notes and handouts; this was frustrating for staff who for sound pedagogic practice had designed interactive, collaborative learning experiences that at least required the learners to attend each class. Such learning could only be experienced, and not passed on through printed notes (Cowan, 1998).

One other significant issue was that the students were also hampered by the fact that they were seeking to learn about professional practices in a working and educational culture that was fundamentally different from that what they had experienced in their prior experiences. For example, students were reluctant to express an opinion in class, and were accepting of the tutor's views, almost in deference to

the teaching team, which was difficult for the staff who at times were looking to provoke challenge and debate for the good of the learning experience in general (Elton, 2008).

After considerable discussion and consultation with a variety of stakeholders, the teaching team concluded that a more holistic response to the needs of the students was required to address the shortcomings identified to date.

### **Employability**

Through examining the desires of the learners, we established that the primary, overwhelming motivation was for the students to obtain at least graduate-level employment in the UK, in a field that was directly related to the programme of study they had chosen. Our responsibility therefore was to ensure that the students were prepared for such employment whilst also demonstrating that they were meeting UK Level 7 learning outcomes (Bloom, 1956).

The teaching team re-visited the module content with a focus on employability skills; however, this was not so much in relation to some of the more generic skills such as preparation for interviews and employer assessment exercises, as these aspects would be taught elsewhere in more general-purpose module content. The focus on employability was related to the skills that an employer might expect from a computing professional, and in the case of this particular module, the skills and knowledge that a professional software designer might have.

This exercise resulted in changes to the proportion of technical content, which was reduced in favour of more experiential learning. For instance, it was felt that sessions that required students to write program code under time constraints, that was then to be passed to different students (acting the role of a client), would help them experience some of the challenges of producing technical work in a pressured environment (Schon, 1983).

Since previous students had found issues around the gathering of client's requirements and generating application designs particularly difficult, the teaching team reasoned that this approach would also begin to address this challenging aspect of the curriculum also.

An additional issue for students was around the QA documentation for the module, which used academic language that was difficult for them to understand and 'constructively align', particularly around 'learning outcome' descriptions (Biggs, 1996).

Continuing with the employability ethos, the teaching team decided to re-write the learning outcomes to make them easier to relate to the story of employability overall. An example of the 'student-friendly' intended learning outcomes are as follows (Brice-Heath, 1983):

*"During this module you will have the opportunity to:*

- *Classify the requirements of a web application by identifying functional and non-functional requirements*
- *Apply the principles of object orientation to design a scalable web application*
- *Analyse the requirements of a system and apply the UML to create a design model of the application*
- *Construct a scalable web application using Object Orientation"*

The content was then arranged in a way that emphasized practice, in an attempt to replicate how a software application designer would learn their craft by actually building the software and testing it. Three case studies were developed for the module in total. The first was used for a running example in the lectures and workshops. A second case study was used for tutorial work. Finally, a separate case study was used for assessment purposes.

The case studies were intended to scaffold student learning so that they could see what aspects of their experience could be transferrable, and what elements would need to be developed for a specific purpose or application (Piaget, 1978). The intention of this was for

students to be able to produce designs and working prototypes of the applications, but not a production-level implementation, as this was beyond the scope of a one semester module.

Students were expected to use these designs and implement applications based on these designs. In practice, students were able to produce working applications that fell short of a fully robust implementation of the design. However, we were preparing students to graduate as programmers rather than as developers.

### **Module evolution**

The first incarnation of the module was delivered as a one hour lecture together with a two hour tutorial, each week for a twelve week semester. As discussed earlier, the emphasis was upon the delivery of technical content; two weeks spent on Object Orientation, with the remaining ten weeks being learning a new programming language and building a working prototype application.

The first change to the delivery was by the conversion of the weekly lecture slot into a workshop session, to allow discussions to be shared amongst students. The tutorials remained an opportunity to cement the foundation concepts discussed in the workshops.

Overall, the balance of the content has changed from an initial two weeks of OO work, to a total of six weeks now. This reflected the need to spend more time supporting the students' acquisition of technical and professional skills, as opposed to concentrating on the delivery of a larger volume of content. The remaining six weeks of the semester is used to focus the students on the processes and experience of implementing a program design.

As the years passed, it became more commonplace for students to arrive with a greater exposure to programming, and the support materials for the module were also refined to reflect this.

## **Enhancing assessment**

Module assessment is comprehensive and has evolved to accommodate both the needs of learners, as well as the demands of employers. Students must be able to design, document and build a prototype application, and this is served through some coursework that includes the production of design documents (use case diagram, class diagram and sequence diagrams), together with program code.

In addition, many employers expect to see potential employees to hold, or be prepared to sit, a vendor-certified examination that demonstrates detailed knowledge of a programming language. Such exams are online, and the teaching team took the opportunity to replicate such an exam by creating smaller online tests that are completed throughout the module delivery. This helps build confidence in the learners, and gives them repeated practice in an online, time-constrained setting. In the case of this module, the online tests were aligned with the Sun Java Programmer Certification test.

The first few cohorts made a distinctive impression upon the teaching team in terms of their assessment attempts. First, the learners had been educated in a culture where examinations were normal and frequent. However, they were not familiar with the style of questioning and examination performance was depressed across the board in almost every case. Second, the learners had no experience of working in groups on summative assessment tasks. The learners made little attempt to differentiate their submissions, leading to large numbers of collusion allegations.

It was clear that the cohort was both bringing new aptitudes to the teaching experience through different educational experiences, but they were also presenting new issues for an assessment method that was tried and tested with Home/EU student cohorts.

This prompted the teaching team to reflect upon the appropriateness of the assessment methods, and a re-think about how the assessment should be designed to assist learning, rather than just attempting to measure and report learning.

While discussing the ‘employability’ ethos, the teaching team concluded that the students’ number one desire – to get a job – should be tackled head-on throughout the module. The students should feel as if they are preparing for a technical job interview, and this would help provide the motivation and engagement that such a scenario brings.

The work that an employer would typically ask to have sight of, usually constitutes a collection of different pieces of evidence that illustrate the capabilities and experience of an interviewee. This collection is a portfolio, and it was decided that this would be the container for a set of assessment activities throughout the module.

Each learner would be required to construct a portfolio that contains the following:

- Project Plan - to help students plan the design and implementation tasks;
- Functional and non-functional requirements;
- User requirement specification;
- Further requirement specification;
- Class diagram;
- Skeleton code.

Each of the students produced a series of incremental artefacts throughout the module, upon which they were given feedback after each submission. The small submissions led to a rapid turnaround of marking, ensuring that students were able to take feedback into account before they work on a subsequent submission.

Maintaining the simulation of a work environment, each submission had to be delivered to an industrial stakeholder, a role that was played by a member of the teaching team. This was to make sure students were not just focusing on grades but actually delivering high quality work.

Week	Lecture	Tutorial	Assessment
1	Module Introduction	Introduction to case study	
2	Design 1 (Functional and Non-functional requirements)	Design 1 (Functional and Non-functional requirements)	Submission 1 (Project Plan)
3	Design 2 (Use case diagrams)	Design 2 (Use case diagrams)	
4	Design 3 (Classes and objects)	Design 3 (Classes and objects)	Submission 2 (Functional requirements)
5	Design 4 (Class diagrams)	Design 4 (Class diagrams)	Submission 3 (Requirement Specification)
6	OO Development using Java 1	OO Development using Java 1	Submission 4 (Further requirement specification)
7	OO Development using Java 2	OO Development using Java 2	
8	Developing JSP pages	Developing JSP pages	Submission 5 (Model Design)
9	Building 2-tier apps with Javabeans and JSPs	Building 2-tier apps with Javabeans and JSPs	
10	Connecting JSP pages to database using JDBC	Connecting JSP pages to database using JDBC	Submission 6 (Skeleton code)
11			Final team assignment submission and online test

**Table 2. Current schedule for Web Application Design and Modelling module.**

The first submission was focused towards a project manager; the second submission was directed towards an end-user; the third and fourth submissions for a team lead; the fifth and last submission was aimed at a development team. Marking schemes contained pertinent reflection questions such as “Is the submission industry ready?”, prompting learners to think more deeply and beyond the assessment task itself. In a number of cases they were encouraged to engage in research outside the module to make sure quality of the work was professional.

### **Soft skills**

Developing ‘soft’ skills is often a challenge for teaching teams involved in the delivery of technical curricula. A particular feature of the ‘professional’ MSc courses was the inclusion of specific content,

delivered via bespoke modules, that exposed students to experiences that they were likely to benefit from.

However, it was noted that student engagement on the professional skills module was low, with learners stating that they “don’t see the point of a professional skills module”.

Clearly, this perception was at odds with employers who were adamant, and unanimous that soft (professional) skills were equally as important as technical abilities.

The need to experience and practice soft skills fitted the re-think of the module, and was a driving force behind selling the ideal of being industry-ready, and using this as a lever to make the content seem more relevant. Sessions that required students to interview a client, or make a case to a team leader, were obvious opportunities to practice group work, communication, negotiation, persuasion and presentation skills.

Some of the tutorial sessions were treated as a miniature project. Learners would be allocated a task and informed as to what outcome was expected of them. Students would then negotiate and agree the time required for the task with their tutor. A designated team leader would then take control of the class. The tutor would leave the classroom at this point after ensuring the students were working together.

At the end of the allocated time the tutor would return to the classroom to check the output delivered from the session. This activity allowed all students on the module to be a team lead at some point. Earlier in the module students would be reluctant to be the ‘team lead’; as the weeks passed, they would gain confidence and students would be more forthright in volunteering to take charge.



**Creating evidence**  
 Throughout the module there will be opportunities to learn and practice new skills.

**Task 1**  
 Wants and needs.  
 What does your customer want? What do they need?  
 What is the difference? Discuss this with a colleague.

**Task 2**  
 It is your first day with a new customer, who has asked for a web application. Your customer has sent these initial requirements "I would like a website which users can use to document their thoughts and maintain an online diary. Visitors to the website should be able to read online diaries. One should be a user to be able to comment on the site."  
 Based on the brief about, what questions should you ask of them in your first interaction with the client?

**Task 3**  
 What could possibly go wrong during your meeting with the customer?  
 Before we can do any analysis of any value we need to be able to capture and express (communicate/present) the requirements as text and graphically.  
 For the textual representation we need to act out the roles that the application will need to accommodate.  
 Once the particular scenarios of use by each role have been described we can start to build a graphical model.

**Task 4**  
 In pairs, write down some functional and non-functional requirements of the application based on your interaction with the client.

**Task 5**  
 In pairs, discuss the different roles that might interact with the application.  
 Write down these roles.

**Task 6**  
 Create one definitive set of requirements and roles for your group.  
 Reflection: How can you evidence your work so far?

**Assignment**  
 For the next session, you should think about the people that you would like to form a group with, and start to explore the balance

**Figure 1. Extract from tutorial notes on university Virtual Learning Environment.**

To cement the ethos of employability (and the associated soft skills practice), the teaching team became more bold in their ideas for innovation within the delivery. Frequent discussions about methods and approaches for embedding employability skills practice into the module led to the use of technology to enhance learning, such as reflective blogs, video diaries, online formative testing, peer marking and feedback delivered through online discussion fora, automated marking rubrics for self-assessment to name but a few (Haigh, 2005).

The content change that had the greatest impact upon students was a decision by the teaching team to use the weekly slot where the entire cohort was addressed, to role play different aspects of a technical job interview. Two members of staff conducted the interview between themselves, while the audience of learners observed. The interview was halted at strategic points to enable questions to be asked, ideas to be explored, and discussions in pairs to be initiated.

The topics that were role-played in the interview then became the mandate for the tutorial session, in which the students were then motivated to engage in the subject material that they had witnessed being discussed and debated in the group setting (Hill, 2011, Tosey, 2002).

Undoubtedly, there was an element of novelty in terms of the style of module delivery, but there was a significant increase in student attendance and engagement after the introduction of this approach. Student feedback was enthusiastically positive, and the module quickly became known for its distinctive delivery.

### **Results**

Twenty separate cohorts have completed this module as part of the professional postgraduate programme over a ten-year period. Whilst not every cohort is comparable in terms of ability at the point of entry onto the course, there is a general trend of improved attainment. There is also a general downward trend in recruitment, and smaller cohort sizes can benefit from an increased proportion of contact from tutors.

Marks for both the small submissions and the online test show the greatest increase overall. The online test is interesting in that when a test is created, there is a considerable effort to establish sufficient questions to populate the bank of questions, upon which random selections are made for the tests at run-time. Over the years, more questions are added which leads to a considerable resource where it is unlikely that there can be any knowledge exchange between prior and present students, as the bank of questions is so vast. Nonetheless, student performance has improved.

While the module has been delivered for some time, the core teaching team has remained the same throughout, with additional staff being drafted in when the cohorts were larger. This has led to a significant amount of specific experience being retained within the module, with significant benefits being that this module has been an

exemplar for the creation of other modules, as well as being a resource for consultation when issues have required diagnosing and resolving.

Year	Cohort no.	Mean mark group work	Mean small submission marks	Mean mark test	No of students	Mean mark overall
2008	1	50	-	53	47	51
2009	1	52	55	61	141	56
2010	1	53	50	61	68	53
2010	2	62	67	57	86	61
2011	1	57	60	57	54	57
2012	1	63	64	45	20	56
2013	1	60	70	48	23	58
2013	2	64	72	51	11	61
2014	1	64	66	51	21	57
2014	2	59	73	48	29	58
2015	1	62	70	59	27	60
2015	2	62	90	60	17	70
2016	1	66	73	57	11	65
2016	2	61	80	63	11	67
2017	1	62	78	65	18	68
2018	1	62	83	79	12	72

**Table 3. Summary of module marks from 2008-2018.**

## Reflection

Using an appreciative inquiry approach to evaluating the module, we observe the following positive outcomes:

- Student engagement with the module improved considerably. Initial delivery of the module reported less than 50% attendance from the third session onwards. Subsequently, this improved considerably with student attendance close to 90% throughout the module.

- Students report (favourably) that the module delivery is structured such that they can incrementally apply their learning as their skills develop, to their coursework assessment. This leads to better distribution of workload across the semester, reduced bunching of assessments, and a better experience for students who have employment alongside their studies.
- The disruptive nature of the module made for an environment where it became easier to introduce teaching innovations. In many cases, the innovations were suggested by the students, which meant that they were accepted if they improved the learning.
- Summative assessment marks have increased over the last four years, and this has been as a result of a steady increase in the quality of the application designs that were submitted. This is likely to have been influenced by the teaching team arriving at a more optimum balance of content versus experience, though as observed earlier, the cohort size has reduced also.
- A minority of students bring prior experience of programming and keeping these students challenged was an important factor of the module. The range of roles that the module offers in terms of practicing professional skills while doing technical work has meant that more able students are being stretched so, the curriculum was able to cater for differentiated learning better.
- Frequent submissions, whilst they were summative, still provided feed-forward for learning. For instance, the construction of use case models in class enabled peer marking to take place. This promoted better conversations around design issues, as the diagram itself does not explain the process too well. As a result, students could learn from others, and apply more advanced topics such as design patterns for example. This facilitated deeper learning to take place in the classroom as students explained their work, enhancing their understanding of the processes (Vygotsky, 1978).
- Personal development planning improved, with much more substantive evidence of soft skills. Frequent opportunities to

discuss learning, and the explicit gathering of evidence – “where is the evidence for your statement?” – fostered an environment where students were actively building their personal portfolios and using this process to self-diagnose gaps in their own learning. Furthermore, they could evidence team working skills in a way that they could not previously (Schon, 1983).

- Alumni reported an appreciation of the role of project management, and how this can, if absent, compromise the quality of design. This is a significant outcome for a module where learners expect to become expert programmers, and has proved important in their ability to perform quickly when they do secure employment.
- Reflective blogging was an opportunity for the students to explore their learning in a way that they were not used to. The result is that they took away considerably enhanced softer skills as well as being more self-aware.
- Reported plagiarism disappeared. The incremental approach to assessment meant that the work had to be developed, and was quality assured checked during the module. A ready-made solution could not be purchased off the shelf.
- Students role-played different occupations - software designer, project manager, business analyst, etc. - and had experience, which helped them decide which job they would like to seek employment in. This gave them an appreciation of the customer supplier relationships that exist within technical environments.
- Students ability to evaluate – the bane of level 7 – was improved also. Regular debates were useful at developing argumentation skills, and the incessant use of public speaking in class, together with relentless prompts to write down and gather evidence, established good study and critical thinking habits within the cohorts.
- The inclusion of a roleplay to set the scene provided some novelty and contributed to an increase in student engagement. It also provided the learners with some experience of what a technical job interview would be like.

- The style of delivery made students pose questions in relation to what they needed to learn as an individual, rather than being directed by a pre-determined curriculum (Tosey, 2002).
- The Intended Learning Outcomes made it clear the relevance of the content to the eventual aim of becoming employable.
- It was an opportunity to introduce questions on a need-to-know basis, without disrupting a more formal schedule of teaching – “I have a question, and get the answer now”.
- The ethos was that we shall find out together – so appreciative enquiry was not just for the delivery team, it was a universal approach to learning and teaching (Seel, 2010).

Whilst the departure from traditional delivery into a more flexible, improvised role play of a technical job interview has yielded positive results, the ability to make the change was underpinned by considerable development work on the module beforehand. The experience of developing a module over an extended period is not always common in the computing domain, where new modules and programmes are being continuously developed. This has a restraining effect upon module development, as there is such an emphasis upon keeping materials current.

However, there were several key factors that enabled the positive results to be achieved.

First, the decision to adopt the ethos of employability was essential, as this enabled the teaching team to coalesce their thoughts around a theme that resonated positively with the students.

Second, the re-writing of learning outcomes, into student-friendly *intended* learning outcomes, was complementary when staff were ‘selling’ the concept of employability and content relevance.

Third, the designing-out of plagiarism from the summative assessment served to prevent academic misconduct from occurring in the first place, but also provided a platform upon which the teaching team could discuss the matter with students.

Finally, the concept of evidence gathering, for the purposes of obtaining a technical job interview, provided an essential scaffold for personal learning and development that was both new and of direct benefit to the students. As such, the introduction of role play rather depends upon the necessary ingredients in place; the prospect of the tutors undergoing a mock interview might be entertaining, but without the support of materials and experiential professional skills of evidence gathering, its potential benefit would be limited.

### References

- Biggs, J. (1996). Enhancing teaching through constructive alignment. *Higher Education*, 32: 347-364, Kluwer Academic Publishers, Netherlands.
- Bloom, B. S. (1956). *Taxonomy of Educational Objectives, Handbook I: The Cognitive Domain*. New York: David McKay Co Inc.
- Brice-Heath, S. (1983). *Ways with Words: Language, Life and Work in Communities and Classrooms*. Cambridge, England: Cambridge University Press.
- Cooperrider, D.L. and Srivastva, S. (1987). Appreciative inquiry in organizational life. *Research in Organizational Change and Development*, 1: 129-169.
- Cowan, J. (1998). *On Becoming an Innovative University Teacher*. Open University Press, Buckingham.
- Erickson, F. (1986). Qualitative methods in research on teaching. Pp. 119-161 in *Handbook of Research on Teaching*. New York: Macmillan.
- Elton, L. (2008). Collegiality and Complexity: Humboldt's Relevance to British Universities Today. *Higher Education Quarterly*, 62: 224–236.

Haigh, N. (2005). Everyday conversation as a context for professional learning and development. *International Journal for Academic Development*, 10(1).

Hill, R., (2011). Risky Business, Educational Developments, SEDA, 12.1, March 2011, ISSN 1469-3267.

Piaget, J. (1978). *Success and Understanding*. Cambridge, MA: Harvard University Press.

Schon, D. (1983). *The reflective practitioner: how professionals think in action*. Boston, Area Publishing.

Seel, R. (2010). *Culture and Complexity: New Insights on Organisational Change*. HEA. <http://tinyurl.com/RichardSeel>

Tosey, P. (2002). *Teaching at the edge of chaos*. LTSN Generic Centre. <http://www.palatine.ac.uk/files/1045.pdf>

Vygotsky, L.S. (1978). *Mind in Society: The Development of the Higher Psychological Processes*. Cambridge, MA: The Harvard University Press. (Originally published 1930, New York: Oxford University Press.)