

Developing Work-Ready Software Engineers using Real-world Team-based Projects as a Catalyst for Learning

Cooper. I M, Gwilliams. C, Ivins. W K, Jones. C M, Turner. M S

School of Computer Science and Informatics,
Cardiff University,
Cardiff, Wales, UK.
CooperIM@cardiff.ac.uk

Abstract—The National Software Academy in Wales is a recently established collaboration between Cardiff University, Welsh Government, the Alacrity Foundation and the technology industry. The aim is to address the gap in the skills market for work ready software engineers. The Academy runs an industry focused BSc degree in Applied Software Engineering that sees local and national industry partners actively participating in the running of the program and gives students the opportunity to work on real-world team-based projects throughout their degree course. This paper reports on the first year of the program: the success and positive engagement from industry partners, the integrated style of delivery across program modules, and the challenges faced going into year 2.

Keywords: Industry engagement; Project based learning; Team Projects; Live projects.

I. INTRODUCTION

In 2013, the then Welsh Minister for Economy, Science and Transport, Edwina Hart, appointed a Business Development Taskforce, chaired by Professor Simon Gibson of Wesley Clover, to consider how the economy of the City of Newport might be improved. The Taskforce concluded that:

“There is a critical gap in the UK between the demand for software developers and the number of skilled people available. We will support the establishment of a ‘software university’ to address this. This will be specifically tailored to create the software developers of tomorrow and is likely to be based on undergraduate and foundation feeder courses.” [1]

National reports have suggested that *“there is a need for 3,100 new entrants a year into IT & Telecoms professional job roles in Wales”* and that *“Software Professionals in Wales show the highest forecast employment growth of all IT & Telecoms occupations at 1.6% per annum”* [2]. The problems facing the industry is that *“Around 40% of digital entrepreneurs say that they face challenges finding skilled digital workers.”* [3], and a number of local and national discussions focus around the “work-readiness” of graduates from computing disciplines. As the 2016 Tech Nation report [3] has noted *“Digital businesses make considerable use of local universities to recruit talent, but entrepreneurs highlight that graduates sometimes lack, either*

the business skills, or the most up-to-date technical skills, needed to go straight into work.” This is in line with a number of the recommendations arising from the 2016 Shadbolt Review of Computer Sciences Degree Accreditation and Graduate Employability [4] which center around improving the work readiness skills of graduates and encouraging universities to tailor degrees to meet the requirements of industry.

Against this backdrop, the idea of a National Software Academy was implemented through the collaboration of Cardiff University, Welsh Government and the Alacrity Foundation. The National Software Academy [5] was developed in early 2014 with the aim of piloting a number of aspects of the concept in academic year 2014-2015 and releasing an industry-focused Applied Software Engineering (ASE) program in academic year 2015-2016. The key aim of the ASE program is to equip students with the skills, knowledge and hands-on experience required to be immediately effective as a commercial software engineer when they graduate.

II. ENGAGEMENT WITH INDUSTRY

Cardiff University has a strong tradition of research-led degrees so the ASE program has taken a new direction with its strong industry focus. It therefore requires a high-level of engagement with industry. Fortunately, the partnership with the Alacrity Foundation provided an excellent entry point to engage with their existing network of contacts via companies they were already engaged with, and to access their industry-active mentors. Welsh Government have also played an important part in promoting the Academy, both to the larger regionally-important companies but also to inward investors, SMEs and to public sector agencies.

A. Pilot year

The pilot year was run with 8 students on a placement year. A key aim was to test the feasibility of running team-based projects and trialing the teaching methods needed to support the delivery of the projects. Students were supported by one full-time academic staff member and one part-time industry-active tutor. Students worked in an office-based environment that resembled a small start-up company.

DOI: 10.5176/2251-3043_5.2.366

The pilot year and the pipeline of projects was managed by a dedicated senior member of staff from the Alacrity Foundation and the pilot year's project manager. Alacrity also provided an existing network of industrial contacts who could be introduced to the Academy concept and used to gauge reaction to some of the aims of the Academy. Trialing assessment methods was out of scope of the pilot year as the placement students were being assessed through their placement module.

B. Established academy

Since the Academy has become established in the local area, it has been successful in generating its own industry contacts. This has been done through a coordinated communications plan and also attending industry-facing events such as the Digital 2015 and Digital 2016 conferences and engaging with sector bodies such as ESTNet (Electronics and Software Technologies Network for Wales). The power of word of mouth should not be underestimated in building networks.

During academic year 2015-2016, the first full year of operation of the Academy, representatives from over 60 different companies have visited the premises or engaged with Academy staff. These have ranged from large multinationals to local startups and SMEs and included both the public and the private sector. It is important to note that not all of these companies have come from the technology sector. A number have come from other sectors including construction, sports/leisure and the charitable sector. However, all have a digital/technological presence or demand which the Academy has been able, or has the potential, to benefit from.

When dealing with such a breadth of different companies and bodies there is a range of different opportunities for companies to access the Academy. These include:

- Provision of projects;
- Mentoring of students;
- Lunch and learn sessions;
- Providing summer placement opportunities;
- Sponsorship;
- Input into curriculum design.

III. PROGRAM DEVELOPMENT

A. Lessons learnt from Pilot year

Key lessons were learnt from the pilot year which have influenced the design of the BSc Applied Software Engineering:

- That group/team-based teaching can be extremely effective in teaching practical skills such as coding, testing, etc. Student feedback has been overwhelmingly positive in respect of this. Small group/team-based teaching can be resource intensive and there is a need to recruit and train appropriate staff who can undertake this form of teaching.
- That a mix of both academic theory and industry practice can be highly effective. The opportunity for students to directly apply theories or concepts to work they are undertaking, either their own websites or designing software for an industrial client, is a powerful teaching and learning tool.

- That there is a real appetite from industry to support the aims and aspirations of the National Software Academy. Without exception, all of the firms who have visited the Software Academy or have held discussions with staff about the Academy have been very supportive and willing to offer assistance.
- That students have found the emphasis on real world working practices such as source control, agile techniques etc. to be beneficial to adding to their portfolio of transferable skills.

B. Underlying principles

With the lessons from the pilot year in mind, the BSc (Hons) Applied Software Engineering was developed through close collaboration of the National Software Academy and The Alacrity Foundation. Cardiff University has a research-led Software Engineering degree, but recruitment was low as most students take the Computer Science degree. The new ASE degree is industry-focused but it incorporates successful elements of the original Software Engineering degree and has similar program learning outcomes.

The program is based at the newly-established National Software Academy in Newport, where the unique ethos places hands-on development using current leading edge commercial tools and techniques coupled with direct industrial involvement at the center of the student experience. With the ASE program being developed in close collaboration with industry (initially via the Alacrity Foundation), and taught by a mixed team with both academic and industrial backgrounds, students learn through the delivery of real software projects; teaming with other students and lecturers in a vibrant 'start-up' atmosphere designed to mirror a real workplace.

The program uses a collaborative project-based learning (PjBL) approach [6] to give students real world experience of working in teams and interacting with industry. PjBL can help students prepare for real-world jobs [6] as it mirrors professional practice [7] where engineers will typically work in team-based projects. Donnelly and Fitzmaurice describe a collaborative project-based learning approach [6] where students apply knowledge and skills to work collaboratively to solve problems that arise in planning, defining purpose, designing, implementing and testing their product. Students are given time to reflect on their learning and evaluate their work.

Danish Universities such as Roskilde University Centre and Aalborg University pioneered wide-scale project-based learning in their engineering programs in the early 1970s [7]. Use of projects in Computer Science and Software Engineering programs is widespread with most universities aligning with project-oriented studies as described by Heitmann [8]. A typical model for UK universities is to have at least one group-based project in the first two years with a significant individual project in the final year. At Cardiff University, our traditional research-led Software Engineering degree provides a team-based project module in each of the three years accounting for 80 out of 360 credits in the program.

The Applied Software Engineering degree at the National Software Academy is unusual as problem-based learning and industry interaction is embedded in all modules. Team-based

projects are used as the catalyst for learning and assessment in all modules in the program, which aligns with a project-organized curriculum described by Heitman [8]. Projects within the program resemble the Problem Project Model described in de-Graaff and Kolmos [7] where students carry out a carefully scoped, full-scale project where the course of action is not planned in detail by the lecturers. Students carry out their projects using current leading-edge commercial tools. Students interact with industry professionals who act as clients for projects on behalf of their companies and/or mentor students as they progress through the program. Students will complete the program with a portfolio of projects that they have worked on during their time at university

At the National Software Academy, the projects aim to represent real industrial experience, which requires the projects to involve work and discussion that spans all of the semester’s module content and learning outcomes. This cross module delivery requires high levels of collaboration and the teaching teams use agile approaches to schedule and deliver teaching and to co-ordinate and scope the projects. The client-based projects are carefully selected by the ASE teaching teams to ensure they provide suitable opportunities for students to achieve the learning outcomes of all modules delivered in each semester.

C. Program Structure

The program emphasizes a range of cloud, mobile, enterprise and web technologies favored in industry - all of which get applied to real projects. This work is used to gain core transferrable skills: everything it takes to work and think like a professional software engineer.

The program is structured as a full-time degree taught over three years with two semesters per year. All modules are 20 credits (200 hours) except for the 40-credit Large Team Project in the final year. There are no optional modules but students will get to choose from a range of different projects in both semesters of their final year. TABLE I shows the program’s list of modules.

Year 1 introduces students to key principles of computing in Computational Thinking and Fundamentals of Computing with Java. These principles are applied in the context of a web development project in the first semester and a mobile development project in the second semester. Students start to develop their professional skills and are introduced to agile working practices in the Software Development Skills modules.

Year 2 gives students the opportunity to work on more complex applications in their projects. Students will gain experience in cloud computing along with various data storage models and solutions. There is a strong emphasis on developing commercial applications so the robustness and integrity of their software is emphasized in the Security and Performance and Scalability modules. Students’ professional skills and understanding are developed through taking responsibility for the management of their projects in Agile Project Management, and appreciating the interaction between development and operations in Developer Operations [9]

TABLE I. MODULES

BSc (Hons) Applied Software Engineering		
Year 1	Year 2	Year 3
Computational Thinking	Database Systems	Commercial Frameworks, Languages & Tools
Introduction to Web Development	Commercial Applications with Java	Emerging Technology
Software Development Skills 1	Agile Project Management	Adopting Technology
Fundamentals of Computing with Java	DevOps	Large Team Project
Mobile Development with Android	Security	
Software Development Skills 2	Performance and Scalability	Managing Software-Enabled Change in Large Organisations

Year 3 will expose students to the latest developments in academic and industry research in the Emerging Technology and Adopting Technology modules and students will be offered a choice of projects in these areas. Their technical skills are expanded by exposure to other commercial languages, frameworks and skills. Students will also take responsibility for developing a large-scale project in the final semester and will be exposed to the challenges of managing change.

Employability skills are embedded throughout the program. Students are shown the importance of good referencing, use of libraries and Web-based information retrieval as a prelude to critical independent study. Oral presentation skills are promoted by class discussions, group exercises and group presentations. Due to the nature of real world projects, legal issues need to be discussed including IP, licensing, and data protection. These employability skills are also practiced and assessed through the projects in line with the module outcomes. The modules and projects encourage the use of current industry standard tools, the choice of which is kept current by continued consultation with industry partners.

The first cohort of the BSc in Applied Software Engineering has just completed its first year. The rest of the paper discusses the approaches used in this first year and reflects on the experiences.

IV. THE NATIONAL SOFTWARE ACADEMY LEARNING MODEL

Teaching at the National Software Academy is placed within the context of experiential learning[10] focusing on the use of projects to provide a realistic grounding for the application of theoretical principals. The work presented to the students evolve from: standard, guided exercises; through small, short, internally-crafted projects; to longer, team-based projects for external clients. The students are required to undertake this work to put concepts in to practice, they are required to conduct self-directed learning and research to enhance this work, which is then analyzed and reflected upon either by discussion, peer

review, presentation to peers, formal written reflection, or summative assessment.

A further notable characteristic of learning in the National Software Academy, is the emphasis on team work. Constructivism theories centre on learners having a framework in their minds. Every encounter that is experienced in the learner's journey will be moulded and fitted into that framework, on the occurrence of more complex encounters, the existing framework may be adapted or, in unfortunate situations, it may collapse leaving the learner to rebuild their framework in a way that the new encounter will fit. Teaching in this style involves aiding the learner to ask and explore appropriate questions. A further step from an individual constructing a framework is the notion of social constructivism [11] which leads on to communities of practice [12] where communities evolve around a common need for education. The team based project work pulls students together both in team based communities and cohort based communities with common education needs. During this project phase of the Applied Software Engineering modules, the emphasis is firmly on the students working in their teams, applying and adapting the basic grounding that has been delivered in a more traditional university style and exploring and expanding around issues that they need to.

Connectivism [13] takes the ideas of Communities of Practice and suggests that learning can be a distributed concept, potentially "residing in non-human appliances"; i.e. a company can learn from its mistakes. In this concept, it is the network – of employees, processes, and documentation - that encompasses the learning that has been achieved. In the ever changing world of software engineering, where our students are being educated to be employed, maintaining the network within a company or specialist field, to enable continued learning, is almost as important as the current knowledge and understanding that an individual employee has.

Within modules on the program there are sessions that focus on both the *relevance* of information and the *quality* of that information that is so available to learners in this digital age. For example: articles on Android development written in 2010, although high quality at the time, may no longer be relevant as the OS has been updated many times since then. This leads to one of the biggest factors in the world of Software Engineering: the ability to remain current in an ever-changing field. This skill must be instilled into the ASE students from early on, and the fact that "*Know-how and know-what are being supplemented with know-where (the understanding of where to find the knowledge needed)*" [13] means that know-where becomes a large part of the educational requirements.

A. Exercise based sessions

While traditional computing-based degree schemes are, typically, taught using lecture theatres, practical sessions and seminars, the Applied Software Engineering degree emulates a small business environment to use the same space for multiple teaching methods.

In Ben-ari[14], the role of constructivist teaching in computer science is explored and the point is made that lectures alone do not allow for deep understanding of complex problems. The ASE program follows this ethos and combines the practical

session with a lecture, allowing students to complete exercises on a concept before moving on.

The outline of the process is this:

- Up-to twenty minutes of theory of a concept with an example to follow
- Students perform an exercise (either individually or in a group) to practice the concept
- Exercises are provided for students to perform self-study outside of learning hours.

This approach does mean that the timetabling becomes an issue as teaching sessions no longer take the typical hour, rather they are scheduled as a full morning or full afternoon of delivery for a module. For the lecturer, this tends to be a more intensive delivery method because of the combination of practical and theory. While the timing of each module does relate to a typical computing-based degree, where practical sessions are typically supervised by postgraduate students. However, following the process outlined above, the lecturer both teaches the theory and supervises the practical session.

As part of the holistic nature of the program, the weekly timetabling is not as fixed as in more traditional programs. The timing of projects, both internal and external, provides checkpoints by which certain content must be covered. Occasionally more of one module's content will be required before a checkpoint than another module's content, which requires the module sessions to be timetabled strategically, and potentially un-evenly, across the semester.

There is also much more flexibility during the project focused parts of the semester, allowing for project specific, and student requested additional content to be delivered in less formal sessions.

As discussed further in Section V, projects make up the bulk of the assessment for all of the modules so, while the delivery can be intensive, the project time that students experience in the latter half of the semester requires very little preparation time; although it is time consuming.

The core focus of this delivery method is that students are taught a concept, practice it immediately and then have the chance to apply it in a team-based software engineering project provided by a real-world client. This not only allows module leaders to oversee individual student progress but also allows learning opportunities for all learning types. Finally, this approach also develops problem-solving skills for students, as they must know which concepts to apply to their team project, when to apply them, and where to search for new, project-specific, techniques.

1) Internal projects

Depending on the structure of the semester, the students may learn through an in-house project for a period. This will typically be very short (two weeks), but it allows the students to practice some key skills (such as source control) before they engage with external clients.

The internal projects are run prior to the client projects and as such, there is reduced contact time dedicated to project work as there is still a high need to deliver new concepts and ideas to the students. Where possible the production of the internal projects is a combination of work developed either in, or guided by, exercise sessions. The advantage of combining the short exercises into a short project is that it gives the opportunity to holistically reflect across module content

These projects also provide a fertile context in which to introduce software lifecycle, project management and general professional skills. The internal projects will require the students to start considering these aspects. Lecturers will lead sessions on topics such as requirements capture, analysis, planning, decomposition, writing acceptance criteria, testing and team responsibilities. These introductions are typically very light. Agile principles are introduced, but there is a conscious effort to ensure students understand the agile principles, and are not just going through the agile processes assuming that this is a 'silver bullet' solution. The teaching effort is more focused on questioning the students as they navigate the project and enabling them to develop their own models of the life-cycle and the effective delivery of value through software.

2) External projects

The ASE Staff spend a great deal of time finding and selecting candidate projects. The highest priority is that the students can achieve their learning outcomes through the execution of the project. While the expectations of the client have to be managed, (i.e. there is no guarantee of a successful delivery of working software at the end of the project) the clients still require something for their efforts. The final delivery will be along the lines of a proof of concept or demonstration model. The value to the client is that they get their ideas interpreted and made real by a set of students within a small time window. This is set against the cost to the client, their time, which typically involves half a day for each meeting. The intellectual property of the work and the code produced by the students are the property of the client, and the students sign non-disclosure agreements to formalize this. The students are also instructed in the principles of intellectual property and the use of external projects adds impact to this instruction.

The 2015-2016 cohort into the ASE program consisted of 23 students. This led to the selection of six teams of three or four students. This number was chosen, as it appeared to offer the greatest opportunity for the project to deliver value to the client, for all team members to make a significant contribution to the overall project outcome and allow for some differentiation in contribution.

For the initial cohort, three external clients were engaged per semester, assigning two project teams to each client, benefiting the clients from having two different interpretations of their problem. The student teams also benefit as, at the end of the project, they can compare and contrast what they have done with at least one other team. The opportunity to see how their peers interpreted the same brief provides a real and valuable learning opportunity and the ASE academic staff facilitate this discussion to draw out the key points.

B. Project case studies

To give an idea of the projects undertaken and the issues surrounding them, one project from each semester is briefly presented.

1) Welsh Football Trust

By the end of the semester, students have the goal of being able to produce a dynamic, client-server web application for an external client. They should be able to demonstrate a competence in HTML, CSS, JavaScript, JQuery and server-side frameworks for data and content serving. The Welsh Football Trust (WFT) came to us with the following problem:

"We (the Welsh Football Trust) have plenty of data on the number of people playing football formally in organized leagues. We do not have data on informal "kick-about". We would like a web-based solution that would allow us to capture the informal playing of football by children in Wales. We would like to know how many children are playing, how old they are, their gender and the location with a view to then planning interventions and incentive schemes to increase the level of football being played."

It is worth noting the level of uncertainty in this brief, and some of the real-world issues that would arise such as:

- How old are the children?
- Can individual details be captured?
- If so, what incentives are possible?
- What level of location data is required?
- Who can see the data?
- What summarization of the data is required?
- How can the site be made appealing to children?

The project ran over six weeks (week 5 to week 11 of a 12 week semester), split into three, two-week iterations with interim meetings in the 7th and 9th weeks of the semester.

The reason that the Welsh Football Trust project (and other similar projects) was chosen was that it had the potential to involve key aspects from each of the semester's modules. It would likely involve aspects of front-end design, usability would be a key feature for the player-facing pages and clarity of data displayed important from the authorized client (WFT) facing pages. Accessibility and responsive design is an important design feature, as the target users would be likely to have a vast range of access devices. The storage of data could be a relatively simple affair, location, date, number of participants, yet the stored data also had the potential to be extended. The client was seeking a solution, not just a website and so, from a requirements gathering point of view this would be a very exciting project.

2) Android projects

By the end of the second semester, the students were to produce an Android application. Alongside this, they learnt more about source control techniques, using an IDE, testing and agile thinking. The Android projects were linked by a common theme; the exploitation of Bluetooth Beacon technology. This technology was brought to the National Software Academy by

local company GCell [15]. GCell were happy to supply a number of beacons to the ASE program as they wanted their technology and their software components to be tested and demonstrated.

One of the clients was a local historical society who required the beacon technology (together with GPS) to provide interactivity to a heritage trail for tourists who wish to commemorate the hundredth anniversary of the end of the First World War. A visitor was encouraged to follow a displayed route, and as they came into proximity with a beacon, historical information, relevant to that location, would be displayed.

For the Android projects there were a range of projects that looked like they could be solved with a very similar framework, this was very convenient from a teaching and management perspective. Each project involved presenting information that was relevant to a specific location, and each project could be integrated with third party functionality, maps for the historical project and social media for others. The other common aspect for the projects was the use and extension of the beacon access API to calculate the closest beacon, and when to notify the user that the closest beacon had changed. All the clients were very open to exploring a range of solution opportunities.

V. ASSESSMENT

During the development of the program, the methods of assessment were at the forefront of the design process. The main aim of the program, being to produce work ready Software Engineering graduates, requires that both practical skills and understanding of theoretical concepts are assessed. Additionally, as project work is predominantly team based, there must be an avenue for students to show off individual skills. There are two main areas of assessment, project based assessment and individual student assessment.

A. Project based assessment

One aspect of live, client-based, project work is that there is the flexibility for the students to explore ideas that are negotiated between the client and student development team. This leads to a challenge at the point of assessment. The assessment must be flexible enough to allow and reward unforeseen direction in the project yet strict enough to ensure academic rigor and allow each student in the team the opportunity to achieve the learning outcomes. In giving students the freedom to explore the project area they also need additional support in the form of enhanced feedback and guidance to ensure that they stay on track from the point of view of the program learning outcomes, and guidance to ensure that the off-piste topics they learn are learnt with due rigor.

As previously discussed, the program is divided into distinct modules inline with UK higher education protocol[16], so the assessment of the students needs to be sufficiently distinct across those modules. Yet if distinct module assignments were set, the size and complexity of those assignments would have to be limited so as to control the student's workload. Setting a cross module project and allowing the modules to assess their distinct parts of the project workload means that generic project tasks do not need to be repeated in multiple assignments. So students have more time to explore more complex tasks in the single

project. They can also gain satisfaction from producing a larger more complete piece of work. Project work assessment within the modules is based on the extraction of module specific content from within the cross module project. This is achieved in two ways:

- a portfolio of submissions, as chosen by the student, detailing **their individual** contributions to the project which demonstrate learning outcomes for the module,
- a team mark for the module content of the project,

1) Portfolio

The portfolio of submissions is graded on the student's ability to design, implement, justify and reflect on the work they undertook. Students are asked to take small aspects of the work that **they themselves** have contributed to the project, present, analyze and reflect on the methods used and discuss alternatives if appropriate. Credit is also given for acknowledgement that they would proceed differently if faced with the same issues again. Students are required to choose work to submit based on demonstrating specific learning outcomes for the module, and word limits are applied to each submission.

The submission side of the assessment also contributes to the formative feedback and reflection throughout the module. A selection of portfolio sections is reviewed part way through the project with the premise of helping the students to prepare for their assessment (which gives them the motivation to prepare sample submissions) but this process also serves as a reflective exercise of the student's work and understanding. The submissions are reviewed as a group of students with a staff member being present for at least one member of the Group's review. This is a scalable process allowing staff to circulate around groups in a larger cohort.

2) Team Marking

The fact that a lot of this program involves team work means that it is very important that the fairness of team mark allocation is correct. There are many papers describing the problems with team project assessment [17], During the primary year, a number of approaches to team mark allocation have been trialed.

In the first semester, the main approach used is to minimize the proportion of the module mark that came directly from the project so as not to cause a huge impact on the overall module grade [18]. For the direct project mark, equal division of the team mark was adopted, with warnings to students if they fail to contribute. This approach worked up to a point as there were only 23 students and staff were able to recognize lack of engagement early on, however, this has been found to be time consuming with the monitoring and extra encouragement of students and so not scalable.

An adaption of this equal division was trialed in the second semester where the portfolio was required to include the project's source control commit log for each student. This allowed the assessment of each individual student's contribution to the project. Students were told in advance that if they were found to have not sufficiently contributed to the project then their marks would be adjusted. This approach of the staff moderating

the individual student's project mark based on individual information is also common [17].

3) *Formative assessment.*

Formative feedback is used within the project for reflection and progression. This is predominantly verbal discussion based. Opportunities for this arise; at client meetings, at sprint meetings, at stand-up meetings, and at technical project team meetings / spin-off technical sessions. This allows for a structured reflective cycle throughout the project phase. Feedback in this style is very focused to the needs of the students, and has the advantage of coming from multiple sources; the client and staff.

B. *Other forms of assessment*

To ensure that students also have the opportunity to show off individual skills, other assessments are also included. Timed programming exercises are used to assess individual programming skills. These are typically a one-day exercise, held in a controlled environment with access to various resources.

Stand-alone assignments are also used to provide specific topics that students work on. This allows the work load to be more distributed across the semester, and feedback from these earlier assignments can be used for reflection throughout the project. Examinations are used to test students' understanding of theoretical concepts.

VI. REFLECTION

A. *Projects*

The real-world projects have provided many challenges for both the students and the staff. First and foremost, the students have been challenged to work as self-organizing teams to meet real client requests for new capabilities. They have had to learn to communicate effectively with people with diverse backgrounds and diverse skill levels in different domains. They have had to learn to plan their work around their continued learning and against defined time-boxing. They have had to learn to share code and other assets, and deal with absences and availability issues.

The ASE staff have also had to adapt as the projects have progressed. Lesson timetables and plans have been adapted on-the-fly according to project needs and some additional guidance has been provided "off-syllabus" to enable the continuation of the project. This has led to spin-out sessions, during the allocated project time, focusing on both generic and project specific content. These sessions work well from a scalability point as student problems can be logged then dealt with in impromptu small group sessions.

1) *Project outcomes and clients*

The cross module nature of the live client project has served to provide a strong motivational feature in the program. Anecdotal evidence has shown that the majority of students have engaged very enthusiastically with the larger team projects, and feedback from the clients involved has been very positive.

In the case of the Welsh Football Trust project, the two teams had prioritized different aspects of the problems. One team had

focused on the individual child and the incentive scheme and was thus capturing individual data, developing skills in data aggregation. The other team had focused on the group of children as a data point and focused on the presentation of the data and on making the web site mobile-friendly. At an interim show and tell, the client realized that he had a decision to make about the data that he wanted to capture and how that would relate to his proposed incentive scheme. The client also became very aware of the child protection and data protection issues that the final project may have to navigate.

In the closing stages of the project, the students had the opportunity to mentor each other, across teams, on the skills that their chosen paths had required. The first team provided mentoring in SQL aggregation (a self-taught element of the code) whilst the second team provided mentoring on the use of the Bootstrap framework and client branding.

In the second semester, the manufacturers of the beacons were extremely satisfied with the project exercise as they had considerable feedback on both their technology and the usability their software. They also acted as clients on behalf of a music venue in Cardiff and were suitably impressed with the ideas that the students had incorporated into the solutions, examples of which include Facebook and Spotify integration.

The historical society mentioned previously is now using the students' work as the basis for an application for Heritage funding to fully develop a visitor app. The students incorporated Google maps, image galleries, and used signal strength (rssi) values to calculate appropriate location choices.

From a practicality point of view, a local council owned heritage site client was slightly less convinced by the beacon apps that were produced for a historic monument. Technically they were good applications but the monument's location and exposure provided certain restraints. The council has shown continued interest in location-based services and the wider application of the technology within the wider council remit. This shows that the client does not need to be 100% satisfied with the project outcome to have gained from the experience.

2) *Project Assessment*

The assessment of the team project has positives and negatives. The use of the portfolio is a good way to introduce rigor, into the assessment process. The distribution of the team mark is still an area that is being worked on for next year. There needs to be a more continual process so that students know how they are progressing throughout the process; this is an area that is discussed more in Section VII.

B. *Delivery*

Cross-module teaching is an unusual format, and relies on a high level of communication and cooperation within the staff semester team.

Scalability in the numbers is being addressed for the coming year. In the first year of delivery, the small numbers meant that it was relatively easy to give personal feedback to individual students in a less organized manner. With more students, a greater number of feedback mechanisms are being planned such as peer review and team marking of formative work.

Cardiff University annually appraises every module that is run via the medium of student feedback forms. These forms quantify the student's opinion how the modules were performed. Modules on the ASE program have received a positive result.

During a first semester focus group, the students were asked for their opinions on the style of delivery. The main issue raised concerned the two external projects run in the first semester, this was considered to be too many because an excessive amount of time was spent in client meetings for a short project – this was agreed and the project load reduced to a single external project in the second semester.

C. Industrial partnerships

Industry engagement throughout the year including mentorship, lunch and learn sessions and provision of projects, has led to a third of our first year students gaining summer placements arranged via the National Software Academy, and others securing related placements independently. Recruitment for these positions has not been “blind” as recruiters have had the opportunity to meet and work with students throughout the year.

The links with industrial partners are continuing, with new and returning representatives continually visiting to discuss various opportunities. One partnership has even lead to a research project, demonstrating the cross-school benefit of these relationships.

D. Industrial Response

Following the first year of the programme, we sent a number of students out on summer placements. These placements were with various industrial partners that we had talked to over the course of setting up the ASE program, and the recruitment was facilitated by staff at the National Software Academy, i.e. encouraged and advertised to both partners and students. Interviewing and contractual arrangements were left to the companies involved. All feedback from the Industrial partners was overwhelmingly positive. With quotes like:

- “StudentA joined [Our] Web Team at the start of the summer and immediately got to work helping us with the project delivery” ... “[The Student] had to learn a new technology quickly (AngularJS) and was assigned a mentor to help [him/her] do so. [The Student] started with some small bug fixes, things like wording or layout changes and then progressed to more complex business functionality. StudentA is a quick learner, enthusiastic and knowledgeable. [The Student] got on with the team and those around [him/her], and took an active role in team meetings and stand ups. As part of [his/her] placement StudentA also produced a tool, initially in HTML, which [they] then converted to a fully fledged AngularJS application. The tool provided the functionality to easily launch the application on multiple environments and through all the entry points/flows and with the correct parameters. [The Student] demonstrated [his/her] progress during the build of the tool to the team through our regular show and tell sessions, in which [they were] confident and communicated well.”

- “[StudentB] is a hard working individual who picks up new skills and software languages quickly. [He/she] has spent 3 productive months working with our core development team. [His/her] experience with the NSA has allowed [him/her] to contribute to our codebase from the outset. [He/She] has produced a number of features that represent valuable additions to the company. [StudentB] exceeded our expectations and we would love to work with [him/her], as well as other NSA students, again in the future.”
- “[CompanyC] has supported the innovative and industry-need based approach that underpins the philosophy of the National Software Academy since its inception.” ... “This summer, we had the opportunity to experience, first-hand, the benefits of the approach, from an employer's perspective, by offering two 12-week summer placements. It was an entirely positive experience. Their familiarity with the agile processes we use and the skills they had gained in extending their knowledge and skills quickly to meet new challenges meant that within the first two weeks it felt as if they were long standing valued, members of the team.”

VII. MOVING FORWARDS

In 2016-17, the ASE program will run the second year modules and also scale the first year cohort to 60 students. As the program moves forward thought is being directed to scalability, both in terms of delivery and assessment, and industrial engagement.

The challenge faced when running team projects is the fairness aspect of individual team members being given grades based on the contribution of other team members. This fairness has to be balanced with the time available to assess each student individually.

The agile / scrum project management style lends itself, structurally, to regular feedback sessions during the sprint meetings with the opportunity to grade students on their contributions for the week / sprint. This meeting would need to be conducted anyway in order to keep track of the student projects. The individual student can be graded on their team play, effort, and technical input. These marks would be a simple low, expected and exceptional and would be accompanied by verbal feedback during the meeting. The combination of the weekly contribution mark will provide a scaling factor when distributing the final team mark.

The potential benefit of this structure to the scalability of the program is that these meetings need to take place any way as a management point of view. Additionally, if staffing time is invested in the first semesters of the course, filling these sheets in within the team environment, then later on peer review of team members, using exactly the same format, can be trialed for a proportion of the sprint meetings. In the same manner that staffing from a project management point of view will be reduced within the sprint meetings as the program progresses.

Whilst the number and the length of projects can vary, there are some general heuristics that are applied:

- The client project should end close to the end of the semester. Typically, this will be the end of the 11th week (in a 12 week semester). This allows the students to have contact time whilst reflecting on the projects.
- The project should have at least three meetings with the external client; a kick-off meeting, at least one interim meeting and a final delivery.

Our current pattern has been for one or two interim meetings, with each meeting being two weeks apart. This means that projects span four to six weeks of the semester (and may span some holiday periods).

VIII. CONCLUSION

The ASE program at the National Software Academy in Wales is a program that shows a novel approach to the extent of engagement between industry partners and the University sector in the generation of degree qualified Software Engineers. This relationship allows the industry, both local and national, to be active and responsive in guiding the direction of the program content in relationship to current employment demands.

The engagement with industrial partners has been exceptionally positive, with over 60 visits in the past year leading to a variety of styles of engagement from talks, student mentoring and provision of student projects with associated guidance. The National Software Academy has also been visited by the Welsh Government's Minister of Economy, Science and Transport, and the UK Government's Director General, Business and Science, Department for Business Innovation and Skills. It gained industry recognition to win the ESTNet (Electronics and Software Technologies Network for Wales) Collaboration of the Year Award [19].

The style of delivery, focusing on learning through doing and reflection, coupled with the emphasis on managing the project and interacting with clients, has prepared the students well for the project phases of the semesters. This has been demonstrated by the positive industry response to the project work delivered.

The range of assessment methods has provided a positive and prolific platform from which to provide feedback to students to help and reflect on their learning throughout each semester. The assessments have also worked to evaluate the learning of the students in a fair and transparent way, producing an expected range of grades similar to other programs in the wider School of Computer Science & Informatics.

One of the primary goals for the ASE degree is to produce graduates who will be highly skilled and ready to contribute to a professional environment immediately following the completion of their degree. Whilst there is not yet a final year cohort to report on, a third of the first year students have participated in industrial summer placements with industry partners who had an awareness of their project work conducted throughout the year. As noted in the paper, the feedback from those placements, both from industry partners and the students, was overwhelmingly positive.

ACKNOWLEDGEMENTS

The National Software Academy would like to acknowledge the particular contribution of John Holvey from the Alacrity Foundation, Prof. Steve Hurley and Dr. Ben Blamey in their work in establishing the NSA.

The National Software Academy would like to thank all our Industrial Partners for their support and enthusiasm. <http://www.cardiff.ac.uk/software-academy/industrial-partnerships>

REFERENCES

- [1] "Newport Business Development Task Force," "City of Newport on the rise: A report to Welsh Government," 2014.
- [2] e-skills uk, "Technology Insights 2012 United Kingdom Northern Ireland," 2012.
- [3] TechCity and Nesta, "Tech Nation 2016: Transforming UK Industries," 2016.
- [4] S. N. Shadbolt, "Shadbolt Review of Computer Sciences Degree Accreditation and Graduate Employability," no. April, 2016.
- [5] "Cardiff University," *The National Software Academy Promotional Video*. UTube, 2015.
- [6] R. Donnelly and M. Fitzmaurice, "Collaborative Project – Based Learning and Problem – Based Learning in Higher Education : Consideration of Tutor and Student Roles in Learner - Focused Strategies," *Emerg. issues Pract. Univ. Learn. Teach.*, pp. 1–12, 2005.
- [7] E. de Graaff and A. Kolmos, "History of problem-based and project-based learning," in *Management of change*, E. de Graaff and A. Kolmos, Eds. Sense Publishers, 2007, pp. 1–8.
- [8] G. Heitmann, "Project-oriented study and project-organized curricula: A brief review of intentions and solutions," *Eur. J. Eng. Educ.*, vol. 21, no. 2, pp. 121–131, 1996.
- [9] M. Httermann, *Dev Opps for developers*. Apress, 2012.
- [10] D. A. Kolb, "Experiential Learning: Experience as The Source of Learning and Development," *Prentice Hall, Inc.*, no. 1984, pp. 20–38, 1984.
- [11] L. S. Vygotsky, "Interaction between learning and development," *Mind and Society*. pp. 79–91, 1978.
- [12] E. Wenger, "Community of Practice: a Brief Introduction," *Learn. doing*, vol. 15, no. 4, pp. 1–7, 1998.
- [13] Siemens George, "Connectivism : A Learning Theory for the Digital

Authors' Profile

- Age,” vol. 2, no. 2005, pp. 1–9, 2012.
- [14] M. Ben-ari, “Constructiveism in Computer Science Education,” *J. Comput. Math. Sci. Teach.*, vol. 20, pp. 45–73, 2001.
- [15] Gcell, “GCell,” 2016. [Online]. Available: <http://gcell.com/>. [Accessed: 05-Jul-2016].
- [16] M. Souto-otero, “Review of credit accumulation and transfer policy and practice in UK higher education,” 2012.
- [17] G. Gibbs, “The assessment of group work : lessons from the literature,” *Assess. Stand. Knowl. Exch.*, pp. 1–17, 1997.
- [18] B. P. Hindle, “The ‘Project’: putting student-controlled, small-group work and transferable skills at the core of a geography course,” *J. Geogr. High. Educ.*, vol. 17, no. 1, pp. 11–20, 1993.
- [19] ESTNET, “Estnet 2016,” 2016. [Online]. Available: <http://www.estnetawards.co.uk/estnet-award-winners-and-finalists-2016/>. [Accessed: 06-Jul-2016].

Dr. Ian Cooper is a Lecturer at Cardiff University’s National Software Academy, his research interests include High Performance and Distributed Computing. He has previously worked for a number of communications companies including Nortel Networks.

Dr. Chris Gwilliams is a lecturer at the National Software Academy and is primarily researching sensor networks, automation and the Internet of Things.

Dr. Wendy Ivins is programme director and lecturer at Cardiff University’s National Software Academy. Her Research interests include software development and management of software development teams.

Mr. Carl Jones is a lecturer at Cardiff University’s National Software Academy. He previously worked for BT for 23 years as a software developer, technical lead and agile coach. His professional interest is in blending the right technologies, methodologies and tools for the problem faced.

Mr. Matthew Turner has been the Manager of Cardiff University’s National Software Academy since June 2014. During that time, he has been responsible for instigating the pilot programme and then coordinating the establishment of the National Software Academy. Prior that he worked in a variety of roles at both Cardiff University and University of Reading.