

Brave New World: Can We Engineer a Better Start for Freshers?

Su White and Les Carr

Learning Technologies Research Group, Electronics and Computer Science, University of Southampton, UK,
Southampton, SO17 1BJ, UK, {saw,lac}@ecs.soton.ac.uk

Abstract - The crucial importance of first experiences in shaping future success has been widely acknowledged. Creating the best foundations in large cohorts of students from diverse backgrounds presents special problems of its own. But a secure foundation can enhance student achievement and improve retention – and the students may even have fun too. Research has suggested that building learning communities can enhance student engagement and achievement.

This paper examines how introducing non-technical activities can establish sound foundations for a university career by a) addressing objectives in the wider curriculum and b) promoting non-technical skills and experience of group working.

A set of changes introduced to five degree cohorts in our academic school were designed to complement enhancements to our technical curriculum introduced during many years of debate and consideration. The changes have impacted upon generic and technical educational experiences. The paper presents an evaluation of the programme of changes through two iterations from the perspective of both faculty and student.

Index Terms – Engineering Education, Group Working, Social Constructivism, Team Skills

INTRODUCTION

The Southampton Experience

The School of Electronics and Computer Science (ECS) at Southampton was awarded the highest possible ratings for excellence in teaching in all our courses by the UK Quality Assurance Agency. A typical first-year cohort will consist of more than 300 undergraduate students studying a range of degree programmes; Electrical Engineering, Electronics, Computer Engineering, Computer Science, and Information Technology. Students are enrolled on either three-year bachelor or four-year masters of engineering degree programmes. Our entry requirements are high and there is intense competition for places. Across the student groups approximately one third of our students will come from outside the United Kingdom, and only 1/10 will be female. Generically across the school we recognised the importance of providing a sound foundation in learning to support students as they begin their undergraduate career.

It is often assumed that ‘Freshers’ Week’ (typically a university-wide welcoming programme) will enable students settle into university life and they will consequently be primed to pursue their degree with gusto. Specially designed induction programmes can be especially helpful to address specific learning objectives [1] and for international students and other groups who may be in a minority (in our case female students are one such group). However discussion in our Academic Committee and Courses Committee which analysed student performance and the general learning experience had also identified a particular need amongst the Computer Science cohort.

Computer Scientists

Our new Computer Science undergraduates are a particularly diverse group in terms of prior experience. Some 25% of the cohort are already highly skilled programmers, while another 25% of the cohort have almost no prior computing experience which raises particular issues in our introductory programming classes. We had already introduced a range of measures, documented elsewhere [2-5], to develop the most effective approach to teaching at this level, however it was still believed that additional work could enhance both the student learning experience and associated student achievement.

In addition faculty had observed that our undergraduate cohort across the school studying Computer Engineering, Electronics or Electrical Engineering appeared to be more socially cohesive than our Computer Scientists. We wondered if perhaps their apparent greater sense of community resulted from the formal structure and particular shared nature of many of their laboratory classes which is not paralleled in the Computer Science degree programme.

The Student Perspective

Material presented is drawn from a range of sources. Students contributed unprompted to the ECS student wiki; colleagues offered unprompted observation; focus group discussions were held with female undergraduates; we also undertook a survey of year 1 undergraduates which elicited 69 completed questionnaires - a return rate of approximately 25%.

Feedback from students in Computer Science, had suggested that a few things could be improved. In focus group discussions the female undergraduates elicited evidence which supported this observation.

Female students report feeling marginal at their first lectures: “it was like they were looking at me and thinking ‘what is she doing here?’.”

Another commented: “Yeah – it might have been useful to have some special programme at the start, I suppose I had got to know the names of half the other students by the end of the second year”.

ISSUES

We were concerned to address the specific needs of one group of students (our Computer Scientists). It seemed that an intervention at the very beginning of their university career which addressed some of the possible gaps in their social experience might be a useful addition to their teaching programme. Consideration of possible solutions for the Computer Scientists led us to speculate whether there was a case for establishing a school wide social introduction to learning.

Existing Practice

In general terms we set out, like any university, to provide a constructive and balanced teaching programme for our undergraduates in ECS. The curriculum is carefully selected and the teaching methods incorporate lectures, tutorials, supervisions and a wide range of practical activities. We place a special emphasis on tutorial support during a student’s first year at university and each student is assigned to a small group which meets on a weekly basis with the academic who is active in the area of their chosen subject specialists.

We are ambitious for our Computer Science students and describe our ambitions to them as follows: “We aim to teach more than just programming. You will be involved in a number of methods of software design and develop the intellectual and computer-based tools needed to support them. Our philosophy is to establish a firm foundation for the study of computer science within a software engineering context.”

Educational Approaches

We are aware of the general literature which point out the ways in which building learning communities can enhance student engagement and improve the outcome of undergraduate studies [6-8]. In our own discipline, establishing the most effective foundations for learning is an ongoing challenge for computer science educators [9, 10]. Some excellent and thoughtful work sought to identify appropriate curriculum and innovative and constructive learning scenarios [11-16].

Educational psychology has been explored, methodologies discussed [10, 17] and numerous courses specifically designed, implemented and assessed which support novice programmers and help students, new to university, build their technical knowledge, skills and understanding. In addition these courses frequently aim to establish sound foundations for approaches to work and study which will prepare the students for their future academic and work careers.

The impact of these innovations has been widely studied and analysed, and their effects (beneficial or otherwise) have been measured and discussed [4, 16, 18, 19]. Within a climate of innovations the commercial success of pair programming has been a strong motivator for the introduction of class and laboratory activities [4, 18, 19]. However, it has been observed, that such social approaches sometimes sit uncomfortably alongside CS students’ previous learning practices which have been seen, in the most part, to be heavily individualistic. Faculty in ECS at the University of Southampton have joined in these debates and designed, refined and redesigned our teaching and then reflected on the efficacy of our changed approaches [2-5].

Our Original Induction

Conventionally in ECS the initial face-to-face components of our student induction have consisted of an extended lecture from the course team which incorporates the distribution of various printed materials, and cursory advice on support and student safety on campus from our senior tutor and the police liaison officer. Later in the week there has been a short programme which includes meeting with the tutor, looking round the computing lab, a wine reception, and a few supplementary lectures on where to find the library and the students’ union.

A number of flaws can be identified in this approach. A large amount of time is spent by academics and support staff talking at students and giving them printed information. Each participant has an important message, but the activity lasts a long time and the students become bored and do not engage with the content being presented. This does not seem to be an ideal first encounter with academic life. Opportunities for social interaction with fellow students are limited and left to happenstance, and are largely reliant upon the initiative of self-confident individuals, who seem to be in the minority. Tutors observed that at initial meetings students (see Figure 1 below) were, not surprisingly, inhibited and the level of social interaction was slow to grow as the term progressed.



FIGURE 1
STUDENTS MEET THEIR TUTOR FOR THE FIRST TIME

The general critique of our induction process was applicable across the entire cohort, however as noted above we remain concerned that we were perhaps not meeting the learning needs of our computer science majors as the term progressed. The challenge was to devise a programme which would address both of these concerns.

TOWARDS A SOLUTION

Given our understanding of the issues we faced we speculated that perhaps changes which expanded student experiences and built capacity for group working and social learning could make an impact on the student experience and the student learning in Computer Science – especially given our observation of some level of absence of social cohesion in this group. At the same time it seemed that such an approach might be helpful in addressing our broader concerns about the shortcomings of our existing student induction processes.

Induction Objectives

So it was that when our department was faced with the question of how to invest in order to enhance the student experience we looked towards introducing non-technical activities which would address objectives in the wider computer science curriculum and promote learning outcomes such as skills and experience of group working. The induction was to be designed to provide a strong practical foundation of general learning skills. These skills will provide essential support for the students in the demanding academic work of their undergraduate program. Our belief was that that by working together students would begin to establish some of the strong social ties which can be an invaluable component of their successful academic career. We explained our objectives to our academic colleagues as follows:

- Teaching them to plan their activities and reflect on their performance, we aim to nurture effective study and working practices from the start.
- Putting them in an environment where they must collaborate together in many different groups, we hope to change the culture of "isolated strugglers"

INDUCTION PROGRAMMES

We devised a pre-teaching programme which we called **JumpStart Student Induction**. Whilst induction programmes of themselves are not new, the particular circumstances of these changes and the large number of students involved means that the topic merits further examination.

The structure of the programme emerged after numerous discussions with colleagues in ECS both informally and through our Academic and Courses Committee Structure and with colleagues across the UK who participate in the community fostered by the Higher Education Academy for Information and Computer Science.

Models Considered

0-7803-9077-6/05/\$20.00 © 2005 IEEE

During our discussions of possible solutions we looked to experience elsewhere. The concept of a student induction is not a new one, although we found no prior example of the particular objectives which we needed to address i.e. encompassing a range of different degree programs along with the special needs of the Computer Science undergraduates. A particular model which seemed it might be appropriate had been evolved by colleagues at Aberystwyth University in Wales [20], However it involved taking a comparatively small group of Computer Science students and involving them in 'outward bound' style activities. In comparison we faced a particular organisational constraint in that our total cohort is large yet we wished to give students an individual experience and to incorporate team learning activities.

Another model, involving team building activities, was operating in other Schools within of our own university and had been run across a range of other similar institutions in the UK. This particularly successful team development programme had been developed some years previously in conjunction with the company BP [21] and was in part designed to establish links between universities and industry. We approached a consultancy firm who specialise in delivering this type of training and asked them if they could help us to develop a compact and portable version of the training for our large student cohort and addressing our own particular needs.

The First Iteration

The classic team training is typically run in very small groups with skilled facilitators. Our large total group size would mean that we would require many facilitators, and there was an additional load associated with the training and orientation of these facilitators.

The facilitators work with groups of around six to eight people, provide them with briefs on their challenges, preserve their activity and intervene if health and safety demands, and then debrief the groups to enable reflective learning. The large group size also created considerable demands with these practical aspects of the program. Where were we to find sufficient space to run the team training with our 300 plus undergraduates? How many briefing rooms did we have available? What would we do if the weather was bad?

To address these particular set of constraints we were forced to schedule the program across two weekends after the very start of term and to run the program on a subsidiary campus of the University different to that where our students were usually taught.

In order to compensate students for this out of hours work so early in their university career, and to recognize the learning which we intended should take place, it was agreed that we should incorporate the group activities as a "course work" in one of the first year modules of study which addresses a wide range of skills which are associated with the professional engineer rather than the details of their chosen subject specialism.

In order to meet some of the objectives of building a community within the discipline groups, students were invited

to attend the programme on a weekend corresponding to their degree specialism.

Before the weekend activities could be run we needed to recruit and train the facilitators over the summer vacation. We advertised amongst our postgraduate students for recruits, and extended our advertisement out of the School in order to identify sufficient helpers. We then ran a series of training sessions for these facilitators enabling them to gain an overview of the activities and to prepare them for the support they were asked to provide.

The first iteration was run on two separate weekends in early October 2003. The weather remained good, participation levels were reasonably high, the food was excellent, the organization was stressful, the translation of the tasks for large numbers (see for example Figure 2) seemed to be successful, the students largely engaged (see Figure 3) and student feedback was largely positive.



FIGURE 2
STUDENTS ATTEMPT A JUMPSTART CHALLENGE.

Outcomes from the event will be discussed in further detail below under the evaluation section, however the immediate observation was that running the event at weekends (it actually clashed with various Rugby World Cup matches which caused some distress to parts of the student body) and after the start of term was not ideal. Clearly an induction needs to be the very first experience – particularly so when it is a being offered for a range of student groups. Over the year we gained feedback from participants, again discussed the options formally and informally with our colleagues and revised the programme for the following academic year.



FIGURE 3
PRESENTING GROUP ACHIEVEMENTS

The Second Iteration

Our first objective in changing the programme was to run it in week zero of term (the ‘Freshers’ Week’ before formal teaching begins). In addition to the original objectives we agreed that we wished to use JumpStart to establish ECS as a primary point of reference for the student. Group activities were designed in which students would participate alongside the rest of their tutor group. The activities were designed to develop a sense of responsibility for the group, which for Computer Science students could be followed up in the organization of shared programming tasks when formal teaching began. At the very simplest level we determined that when any student went to their first lecture they would be able to look around and identify at least one other person whom they already knew.

The initial activities were presented as a set of team challenges in the form of a scavenger hunt, they were designed to be related to things the students needed to know to do course and be student; in ECS; at the University of Southampton; within the city of Southampton. Teams were given disposable cameras to capture their answers and create a poster. Some challenges were paper based designed to encourage them to work their way through some of the course handbook and understand rules and regulations which we considered to be particularly important.

Rather than having post graduate facilitators we recruited student helpers from existing second years – students who had participated in JumpStart version 1 during the previous year.

We still believed in the value of the more explicit team training components which we had included in JumpStart 1, however we felt it was more straightforward to schedule these activities for another part of the year.



FIGURE 4
STUDENTS AT THE JUMPSTART PRIZEGIVING.

FEEDBACK AND EVALUATION

In the first year of JumpStart alongside the positive feedback we received reasonable criticism that the program had come insufficiently early in the year and that the weekend scheduling was far from ideal. As organisers we agreed with these views and accordingly moved the events to week zero.

Once again, there were positive comments as is evidenced by this student contribution to the undergraduate Wiki

“Jumpstart was a great program to give an advantage over other schools who do not have a similar program like this..... all the tasks were well worth doing as you get to know where things are in uni and the city.”

However, this time, rather than resenting losing a weekend, some participants felt that it had interfered with their opportunity to take part in other university wide or hall based activities.

Part of the challenge had been for students to schedule their own time and clearly this had caused a number of them problems. Similarly it had been our explicit objective to engineer the social context so that students were more likely to spend time with other undergraduates also studying in ECS. We have observed that our engineering programs are often more time-consuming for students than other degree courses and we believe that close identity with a fellow students on the same program can support students in successfully sticking to the workload we placed upon them. For these reasons we were happy to live with the criticisms, particularly so because of comments from numbers of students who clearly appreciated the opportunity for social contact with others in ECS. A selection of those comments collected during an extensive survey of the first year initial experience are shown below

“Jumpstart was good at it gave me stuff to do in Freshers week.”

“JumpStart was good to let us meet our fellow students.”

“Friendly atmosphere, made everyone feel welcome.”

We can only speculate if these are comments that have come from students whom we would otherwise have identified as our ‘isolated strugglers’.

Academic views

From the perspective of our academics, a number of observations have been made.

“I don’t know what you did to my tutees, but this week they are actually talking!” (unprompted feedback from one academic after the following week’s tutorial.)

At the poster presentation and prizegiving there was a palpable difference (see Figure 4) compared to previous years ‘receptions’ with students chatting animatedly to each other rather than standing in embarrassed silence amongst the academics.

Another colleague reported excitedly of how in week four his tutor group had arrived expressing concern that one of their number had not been attending that week and they informed the tutor that they intended to go and find the student and see if they could help him.

Senior students views

The decision to use second-year students that had experienced the initial JumpStart course had a number of beneficial outcomes. Firstly, the buy-in from the undergraduate student helpers was much greater than had been apparent from the research students in the previous year. Secondly, the helpers have become motivated to start an ‘ecumenical’ ECS student association which caters for students from all degree streams. Traditionally, all such activity has been segregated by degree stream (software engineers vs hardware engineers, reflecting the split origins of the school almost 20 years ago) despite the fact that hardware-oriented degree students also learn programming and software-oriented students must take electronics lab classes.

FURTHER WORK

As well as continuing to refine our induction programme, the work on enhancing our earliest programming modules continues. We intend to draw together analysis of both these experiences for further consideration. As we follow our students through their entire university career we will be able to make some further analysis as to the total impact of this important change.

CONCLUSIONS

We introduced a number of changes motivated by a wide range of evidence that as an academic school we needed to enhance the initial support which we provided for our undergraduates.

The introduction of these changes posed logistical challenges because of the large numbers we were dealing with. In addition we were addressing objectives which would impact on both the generic set of student skills and specific technical agendas within computer science.

In introducing these changes we observed benefits which have been manifest in terms of the social climate which exists

amongst undergraduates within the school. Student feedback has evidenced some of these benefits, and we are also seeing a burgeoning of less formal social activities amongst our undergraduate students.

It is important to acknowledge resistance to our intentions which we experience from some of the students. The difference in the student perspective and the academic perspective shows up the cleavage which exists between a short term and a long-term view. As academics who have regularly attended welcoming meetings with students and sat through interminable tutorial meetings we have a clear belief in the value of a programs such as JumpStart. We have demonstrated that it is possible to run school-wide inductions on a large-scale and we continue to observe our computer scientists with interest in the belief that it is possible to enrich their learning through a greater emphasis on social activity.

ACKNOWLEDGEMENTS

I should like to acknowledge the support of colleague from across Electronics and Computer Science who have contributed in various ways to this scheme, particularly Andy Gravell, Denis Nicole, Eric Cooke Hugh Davis and Joyce Lewis. Special thanks are also due to Mike Rawlins of Chalybeate who assisted in the design of the first incarnation of JumpStart, and of course to our postgraduate student helpers and our willing undergraduate participants and helpers in the second round.

REFERENCES

- [1] N. S. Edward, "Evaluation of a Constructivist Approach to Student Induction in Relation to Students' Learning Styles," *European Journal Of Engineering Education*, vol. 26, pp. 429-440, 2001.
- [2] H. C. Davis, L. A. Carr, E. C. Cooke, and S. A. White, "Managing Diversity: Experiences Teaching Programming Principles.," presented at the 2nd LTSN-ICS Annual Conference, London, 2001.
- [3] L. A. Carr, Davis, H. C. and White, S. A., "AnnAnn - a Tool to Scaffold Learning about Programs," presented at the Fifth Annual Conference of LTSN-ICS, Ulster, UK., 2004.
- [4] G. B. Wills, Davis, H. C. and Cooke, E. C., "Paired Programming for Non-Computing Students," presented at the Fifth Annual LTSN-ICS Conference, Ulster, UK., 2004.
- [5] H. C. Davis, Carr, L. A. and White, S. A., "Conducting Examinations On-line," presented at the 11th International Conference ALT-C, Exeter, UK, 2004.
- [6] National Survey of Student Engagement, "Improving the college experience: National benchmarks of effective educational practice," Indiana University Center for Postsecondary Research, Bloomington, IN 2000.
- [7] C.-M. Zhao and G. D. Kuh, "Adding Value: Learning Communities and Student Engagement," *Research in Higher Education*, vol. 45, pp. 115 - 138, 2004.
- [8] J. S. Brown, A. Collins, and P. Duguid, "Situated cognition and the culture of learning," *Educational Researcher*, vol. 18, pp. 32-42, 1989.
- [9] P. B. Henderson, "Anatomy of an introductory computer science course," presented at The seventeenth SIGCSE technical symposium on Computer Science Education 1986, Cincinnati, Ohio, United States, 1986.
- [10] R. E. Mayer, "The Psychology of How Novices Learn Computer Programming," *ACM Computing Surveys*, vol. 13, pp. 121-141, 1981.
- [11] R. Shackelford, L. Cassel, J. Cross, J. Impagliazzo, E. Lawson, R. LeBlanc, A. McGettrick, R. Sloan, and H. Top, "Computing

- curricula 2004: the overview project," presented at The 35th SIGCSE technical symposium on Computer Science Education, Norfolk, Virginia, USA, 2004.
- [12] E. B. Koffman, P. L. Miller, and C. E. Wardle, "Recommended curriculum for CS1, 1984," *Communications of the ACM*, vol. 27, pp. 998-1001, 1984.
- [13] R. H. Austing, B. H. Barnes, D. T. Bonnette, G. L. Engel, and G. Stokes, "Curriculum '78: recommendations for the undergraduate program in computer science — a report of the ACM curriculum committee on computer science," *Communications of the ACM*, vol. 22, pp. 147-166, 1979.
- [14] ACM, "Computing curricula 2001," *Journal of Educational Resources in Computing*, vol. 1, 2001.
- [15] M. Samaka, "Changing a computer science curriculum in light of computing curricula 2001," *SIGCSE Bull.*, vol. 34, pp. 32--35, 2002.
- [16] M. van Veen, F. Mulder, and K. Lemmen, "What is lacking in curriculum schemes for computing/informatics?" in *Proceedings of the 9th annual SIGCSE conference on Innovation and technology in computer science education*: ACM Press, 2004, pp. 186--190.
- [17] Y. Magrass, "Computer science curriculum: technography, technocracy, technology, or theology?" *ACM SIGCSE Bulletin*, vol. 17, pp. 59-64, 1985.
- [18] T. VanDeGrift, "Coupling Pair Programming and Writing: Learning About Students' Perceptions and Processes," presented at Proceedings of the 35th SIGCSE technical symposium on Computer Science, Norfolk, Virginia, USA, 2004.
- [19] L. A. Williams and R. R. Kessler, "All I really need to know about pair programming I learned in kindergarten," *Communications of the ACM*, vol. 43, pp. 108-114, 2000.
- [20] M. Ratcliffe, J. Woodbury, and L. Thomas, "Improving Motivation and Performance through Personal Development in Large Introductory Software Engineering Courses," presented at Conference on software engineering education and training, Covington, KY, 2002.
- [21] E. Dunne and M. Rawlins, "Bridging the Gap Between Industry and Higher Education: Training Academics to Promote Student Teamwork," *Innovations In Education And Training International*, vol. 37, pp. 361-371, 2000.