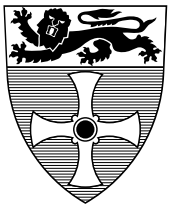
UNIVERSITY OF
NEWCASTLE



University of Newcastle upon Tyne

COMPUTING SCIENCE

Active Learning in Computing: Engaging Learners in a Cross-Site Team Project

M. Devlin, L. Marshall, C. Phillips.

TECHNICAL REPORT SERIES

No. CS-TR-997 January, 2007

TECHNICAL REPORT SERIES

No. CS-TR-997 January, 2007

Active Learning in Computing: Engaging Learners in a Cross-Site Team Project

Marie Devlin, Lindsay Marshall, Chris Phillips.

Abstract

Active Learning in Computing (ALiC) is a CETL project led by the University of Durham, with the University of Newcastle, Leeds Metropolitan University and the University of Leeds as partners. The project focuses on increasing the level of student engagement within the Computing curriculum, aiming to make the student experience more relevant to industry. This paper concerns an experiment being undertaken jointly by the ALiC researchers at Newcastle and Durham.

The pedagogical aims are to increase engagement and simulate the working practices of large companies, specifically running software development projects with teams at different geographical locations. The experiment consists of an assignment shared between teams of second year students on the Software Engineering Modules at Durham and Newcastle, where regular communication helps achieve success. The teams can use communication methods such as email and face-to-face meetings on-site and email, SMS, bulletin boards and video-conferencing for contacting their off-site counterparts. The structure of the assignment gives an insight into Software Engineering in an industrial context, makes problem-solving more realistic and also encourages the development of transferable skills. The assessments allow students to reflect upon their individual and team performance and to explore the roles and skills required when working in teams. This paper reviews the Newcastle students' perceptions of their skill levels and choice of team structure at the beginning of the team project and compares them to their reports at the interim stage. It also outlines what the module leaders have learned so far and how they will refine the design of the assignment in the future.

Printed and published by the University of Newcastle upon Tyne,
Computing Science, Claremont Tower, Claremont Road,
Newcastle upon Tyne, NE1 7RU, England.

Bibliographical details

DEVLIN, M., MARSHALL, L., PHILLIPS, C..

Active Learning in Computing: Engaging Learners in a Cross-Site Team Project
[By] M. Devlin, L. Marshall, C. Phillips.

Newcastle upon Tyne: University of Newcastle upon Tyne: Computing Science, 2007.

(University of Newcastle upon Tyne, Computing Science, Technical Report Series, No. CS-TR-997)

Added entries

UNIVERSITY OF NEWCASTLE UPON TYNE
Computing Science. Technical Report Series. CS-TR-997

Abstract

Active Learning in Computing (ALiC) is a CETL project led by the University of Durham, with the University of Newcastle, Leeds Metropolitan University and the University of Leeds as partners. The project focuses on increasing the level of student engagement within the Computing curriculum, aiming to make the student experience more relevant to industry. This paper concerns an experiment being undertaken jointly by the ALiC researchers at Newcastle and Durham.

The pedagogical aims are to increase engagement and simulate the working practices of large companies, specifically running software development projects with teams at different geographical locations. The experiment consists of an assignment shared between teams of second year students on the Software Engineering Modules at Durham and Newcastle, where regular communication helps achieve success. The teams can use communication methods such as email and face-to-face meetings on-site and email, SMS, bulletin boards and video-conferencing for contacting their off-site counterparts. The structure of the assignment gives an insight into Software Engineering in an industrial context, makes problem-solving more realistic and also encourages the development of transferable skills. The assessments allow students to reflect upon their individual and team performance and to explore the roles and skills required when working in teams. This paper reviews the Newcastle students' perceptions of their skill levels and choice of team structure at the beginning of the team project and compares them to their reports at the interim stage. It also outlines what the module leaders have learned so far and how they will refine the design of the assignment in the future.

About the author

Marie Devlin is a teaching fellow for the Centre for Excellence in Teaching and Learning - Active Learning in Computing.

Lindsay Marshall is a Senior Lecturer for the School of Computing Science and a founding member of the Centre for Excellence in Teaching and Learning - Active Learning in Computing.

Chris Phillips acts as Director of Teaching for the School of Computing Science and is Degree Programme Director for undergraduate programmes in Computing Science, Computing Science (Software Engineering), and Information Systems. He is a founding member of the Centre for Excellence in Teaching and Learning - Active Learning in Computing.

Suggested keywords

ACTIVE LEARNING,
STUDENT ENGAGEMENT,
COMPUTING CURRICULUM,
SKILLS FOR INDUSTRY

Active Learning in Computing: Engaging Learners in a Cross-Site Team Project

Miss Marie Devlin, Dr Lindsay Marshall, Dr Chris Phillips
Marie.Devlin@ncl.ac.uk, Lindsay.Marshall@ncl.ac.uk, Chris.Phillips@ncl.ac.uk

Abstract

Active Learning in Computing (ALiC) is a CETL project led by the University of Durham, with the University of Newcastle, Leeds Metropolitan University and the University of Leeds as partners. The project focuses on increasing the level of student engagement within the Computing curriculum, aiming to make the student experience more relevant to industry. This paper concerns an experiment being undertaken jointly by the ALiC researchers at Newcastle and Durham.

The pedagogical aims are to increase engagement and simulate the working practices of large companies, specifically running software development projects with teams at different geographical locations. The experiment consists of an assignment shared between teams of second year students on the Software Engineering Modules at Durham and Newcastle, where regular communication helps achieve success. The teams can use communication methods such as email and face-to-face meetings on-site and email, SMS, bulletin boards and video-conferencing for contacting their off-site counterparts. The structure of the assignment gives an insight into Software Engineering in an industrial context, makes problem-solving more realistic and also encourages the development of transferable skills. The assessments allow students to reflect upon their individual and team performance and to explore the roles and skills required when working in teams. This paper reviews the Newcastle students' perceptions of their skill levels and choice of team structure at the beginning of the team project and compares them to their reports at the interim stage. It also outlines what the module leaders have learned so far and how they will refine the design of the assignment in the future.

Introduction

Active learning in Computing, (AliC), is a CETL project led by the University of Durham, with the University of Newcastle, Leeds Metropolitan University and the University of Leeds as partners.

AliC promotes the development of a more cohesive Computer Science curriculum where exercises and assessments help develop skills that are more significant to the requirements of employers. This curriculum places emphasis on non-technical skills such as teamworking, leadership, communication etc. as well as the traditional technical skills such as programming.

This paper describes an experiment being undertaken jointly by AliC researchers at Newcastle and Durham that encompasses the development and assessment of all these skills.

The first part of the paper describes the experiment and outlines the pedagogical motivations and aims behind it. Section two reviews the students' perceptions of their skill levels and choice of team structure at the beginning of the experiment and

compares them to their reports at the interim stage. This section analyses the results and their implications for teaching and learning.

The next (fix when finished) section focuses on the technologies used to facilitate aspects of the experiment, their impact on the learning experience and implications for further running of the experiment. The last (ditto)section outlines what the module leaders have learned so far and how the design of the experiment will be refined in future.

1. The Software Engineering Team Project

The Software Engineering Team Project at Newcastle is a compulsory second year module that runs within the School of Computing Science every year for the whole academic year. Students from the majority of programmes take the module.

There is no exam for the module at Newcastle and assessment is based solely on continuous assessment of the various deliverables. The deliverables produced during the first 5 weeks deal with lecture material and come under the 'Software Engineering' component and contribute 25% of the module mark. A small portion of marks is allocated to some 'external' activities such as interviews. The remaining deliverables come under the 'Team Project' component and contribute approximately 75% of the module mark.

Team marks are weighted using two self-assessment percentages returned by each team to derive an individual's proportion of the team mark. [4]

Normally students work with peers from their year group at Newcastle.

In conjunction with researchers from Durham, the Team Project for 2005/2006 was designed so that students from both sites had to collaborate with each other in order to achieve the project objectives.

The task was to design a holiday guide application for a fictional holiday company 'Haway Holidays'. The application would be used by holiday makers as a 'digital assistant' tailored to their particular holiday needs and would be loaded up by Haway's staff before dispatch to the customer with any information that might be relevant or of interest to that customer and the particular location they were visiting. Newcastle based teams had to develop the software for a PDA whereas Durham-based teams had to develop for a mobile phone prototype.

Each Newcastle team consisted of 6-7 members and had a counterpart team of 4-6 at Durham. There were 12 teams at each site. The teams had to work together to construct commonalities in their prototype systems with regards to 'look and feel' in order to develop a 'corporate identity' with their cross-site counterparts.

The module leaders had to bear in mind that students could not be penalised if there was poor communication between teams or the experimental part of the module failed. It was agreed that the collaboration would be worth a minimal part of the total marks to be awarded for the project.

Newcastle teams were told they should hold at least one formal meeting a week with their on site members and that contact with their Durham counterparts could take place using any communication means at their disposal including email, phone, in person or video conferencing.

1.1 Cross Site Differences

There are some differences in how the Software Engineering module is delivered at Newcastle and Durham and module leaders had to ensure that learning outcomes and assessment methods were not compromised during the experiment.

At Newcastle students on the module have 5 weeks worth of lectures at the start of term. These lectures deal with the background material on Software Engineering that they need for the project e.g. Software Lifecycle and Process Models, Requirements Analysis, Design and Configuration Management, Testing and Debugging etc.

During the first 5 weeks of the course Newcastle students also have two one-hour practical sessions in which they can practice the techniques shown in class or do coursework etc. From Week 6 onwards Newcastle teams could arrange meetings and work on their project at any time suitable to themselves and not just the scheduled practical time. Practical classes were not staffed.

Durham's timetable was much more structured with 7 lectures from the beginning of term and formal practicals each week in which to complete their projects as well as other times of their own choosing.

There were also differences between sites with regard to the schedule of deliverables. Newcastle students had small team and/or individual deliverables almost every week whereas Durham students had larger deliverables with longer periods between deadlines throughout the project duration.

Another difference was the use of **'**monitors' at Newcastle - these are members of staff whose role is to observe the formal team meetings and to answer questions. The monitors are not responsible for managing or directing the project in any way. This is in contrast to the role of third year students at Durham who act as project managers for the teams and whose participation is assessed.

2. The Use of Video Conferencing

The aim of the ALiC team in incorporating video-conferencing and cross-site working into the Team Project design was to engage students more actively in the learning process and to emulate what would really happen in industry. The majority of students undertake group work throughout their degree programme. However, problems designed for group work are often within 'toy' domains and have little or no relation to problems faced by large companies. Module leaders wanted to make the project realistic.

Running a software development project between teams at different geographical locations via video conferencing technology made the project more realistic because cross-site development between teams at a geographical distance has become more commonplace in industry and improved communications technology has made it easier and more cost-effective than travelling.

The cross-site teams experienced some technical difficulties at the start of the first semester just getting used to the technology. Students were frustrated when they could not hear the audio properly or could not get the cameras to give a good picture of their cross-site team-mates. There were problems getting the video and audio from each of the sites on occasion e.g. Newcastle had a virus on one of the video machines and this made it really slow, the Manchester AG Bridge went down on one occasion and meetings had to be rescheduled etc. Students from both sites also found it difficult to schedule the meetings. This was compounded by the fact that the Newcastle facility had to be staffed whereas Durham students were in control of their facility and had to set it up themselves for each meeting. Newcastle students had to work around the schedule of the staff member and also the fact that the room could be booked by other groups within the School. Durham students had priority booking for their room and therefore were only restricted by their timetable.

Teams found it frustrating if things went wrong at the time of the actual meeting and they tended to blame the other team if their meetings had to be rescheduled. They found the differing deliverables and differences in schedule confusing. Scheduling

was also difficult because students from the onsite teams were on different courses and found it hard to organise themselves never mind coordinate with team members from a different site.

Many teams chose to keep email as their primary communication method but still experienced problems sending attachments on both sides due to mailbox limitations and also with people simply not checking their emails on a regular basis.

Students also found it difficult to view their Durham counterparts as part of their team and video conferences were approached with a certain wariness, their language became formal and restricted. They generally found it hard to be themselves and were initially shy about talking to each other but this changed as they became more familiar with the technology and each other. In general the communication and problems highlighted their lack of experience with regard to running a project which is precisely what they were there to learn. Over time the teams generally learned to manage their time better, to be more precise in their communications with each other and to act in a professional manner. There were some problems with teams who did not communicate with each other regularly enough or well and these teams had to be reminded by staff to keep up communications. Staff tried to ensure that difficulties were resolved. Some teams also had problems with team mates on their own site e.g. non-attenders, some students not pulling their weight or not participating fully, some members being more dominant, general everyday team troubles that were resolved between themselves or with the help of their monitor or the module leaders.

On a more positive note, students reported that they liked the video conferencing technology and found it interesting and different. They liked the idea that they were getting experience and also taking part in something totally different in the module than students had done before.

3. Recognising & Developing Skills

Another motivation for the experimental design of the Software Team Project is that Computing students tend to concentrate on their technical skills at interview and are not confident in discussing their communication, teamworking, planning and project management skills or their ability to adapt and solve problems and to deal with other people which is what employers are really seeking.

The Association of Graduate Recruiters (AGR) suggests that subject choice is less important than evidence of broader skills when employers are trawling for new talent. They are not finding what they want. According to the AGR, almost half of Britain's top companies did not expect to receive "sufficient applications from graduates with the correct skills". In particular, employers found difficulty recruiting graduates with "softer skills such as team-working, cultural awareness, leadership and communication skills." [7] If a graduate has an aptitude for programming their skills can be developed by a company over time. Most companies do not expect expert programmers on graduation. They do expect a more ‘rounded’ individual i.e. a graduate who has “developed as a person and acquired a range of intellectual qualities so as to be capable of performing in an intelligent way outside the confines of what has been taught in formal courses”. [2]

One of the most important tasks that students at Newcastle undertook during the first five weeks of the module was to review their skills, figure out what role they could best play in a team and at what stage in the software engineering process their particular skills and strengths could be used.

Students were given a Self-Assessment Tick list (Figure 1), based on Belbin team roles. [5] Belbin team roles “describe a pattern of behaviour that characterises one person’s behaviour in relationship to another in facilitating the progress of a team.” [6]

The tick list outlines a list of roles and gives descriptions of the typical skill strengths and behavioural characteristics associated with each particular role. Students were asked to indicate whether they felt they possessed the skills and strengths of any of the roles and also to identify whether they believed these characteristics were a personal primary or secondary strength.

Team Roles	Primary Strength	Secondary Strength
Innovator - produces ideas, imaginative, unorthodox, radical, clever, uninhibited. (Can be over-sensitive, prickly. May need careful handling.)		
Investigator - Finds things out, always knows someone who…, brings information back to the team, enthusiastic, gregarious. (Can be lazy and complacent.)		
Chair - Self-confident, commands respect, good speaker, thinks positively, good at guiding team. (Can be domineering, bossy.)		
Shaper - Energetic, drives everyone along, needs to succeed, make things happen. (Can be disruptive and argumentative, impatient and a problem if things don't go their way.)		
Evaluator - Careful, makes intelligent judgments, tests out ideas, evaluates proposals, helps the team avoid mistakes. (Can become isolated and aloof, pessimistic or over-critical).		
Teamworker - sympathetic, understanding, sensitive, shows a strong concern for social interaction, leads from behind. Places the team above personal concerns. (May be indecisive).		
Organiser - Methodical, hard-working, reliable, orthodox, turns idea into plans which are feasible and gets down to tasks that need doing. (Can be inflexible and uninspiring).		
Finisher - Painstaking, conscientious, follows through and works hard to finish things properly. Meets deadlines and pays attention to detail. (Can be over-anxious and perfectionist).		

Figure 1: The Self Assessment Tick List

There were a total of 83 Newcastle students registered for the module 2005/2006 and 83 submissions for the skills assessment task.

Role/Skill	Primary	Secondary	Primary %	Secondary Skill %
Innovator	30	45	36	54
Investigator	32	39	38.5	47
Chair	18	35	22	42
Shaper	23	31	28	37
Evaluator	45	31	54	37

Teamworker	47	31	57	37
Organiser	39	32	47	38.5
Finisher	35	32	42	38.5

Table 1: Students' view of their skills at start of project

As can be seen from the figures in table 1 very few of the students thought they had leadership qualities as a primary skill at the start of the project. The roles of Chair and Shaper scored very low, just 22% and 28% respectively for primary skills and for both primary and secondary skills combined they had the least ticks with 35/36% of students indicating they thought they did not have this skill at all. The students who did think they had leadership skills made reference to previous experiences in leadership with regard to hobbies and previous work experience.

The majority of students (94%), thought that they had the skills of a teamworker - which is good but it is also a fairly non-specific/generic category and rather a 'safe' option. On completion of the tick list students discussed their skills and many referred to the fact that they were not sure what skills they possessed because they had not undertaken such a large project before and were not sure where they fitted in.

The Evaluator skills also scored fairly high among students as a primary skill. Quite a few of the students commented on their lack of Organiser skills and how they would like to improve them or in some cases avoid the responsibility of such roles as they felt they would let the team down. A majority of students thought that they possessed the skills of the Innovator. Module leaders felt that this was because innovation is a skill most likely associated with technical ability and programming. This idea seems to be borne out in the role preference results. In the role preference section of the assignment, students were asked to indicate which role within the project they felt their strengths would be best suited to.

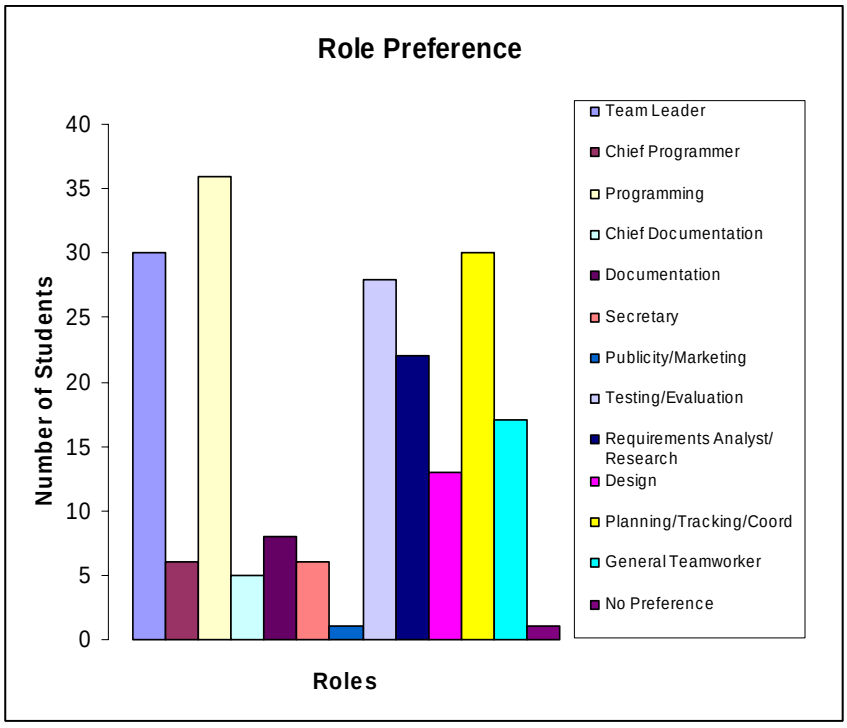


Figure 2: Role Preferences at Start of Project

On average students indicated 3 preferences with regard to the role they played in the team. A high percentage of students chose a programming role, (43%), possibly because that is the skill they most readily associate with a Computing Science graduate and the skill most likely to get them a job when they leave university.

The main reasons given for choosing programming role were that this was the role they felt most comfortable in. Some students, especially Information Systems students who indicated this preference gave the reason that they wanted to improve their programming skills during the project. The Information Systems course at Newcastle does not have a hard emphasis on programming and students on this course take fewer programming modules than students on the Computing Science course.

A good number of students indicated that they would like a team leader position, (36%). Students who chose this option said their motivation was mainly to gain experience of this role, improve confidence and find out what it was like.

One of things commented on by some students who chose this role was that they thought the project was a 'safe' testing ground and if mistakes were made the consequences would not be as dramatic as they would be in the real world. They indicated that this aspect meant they felt quite comfortable in choosing a role outside their comfort zone in which they could test and improve their skills.

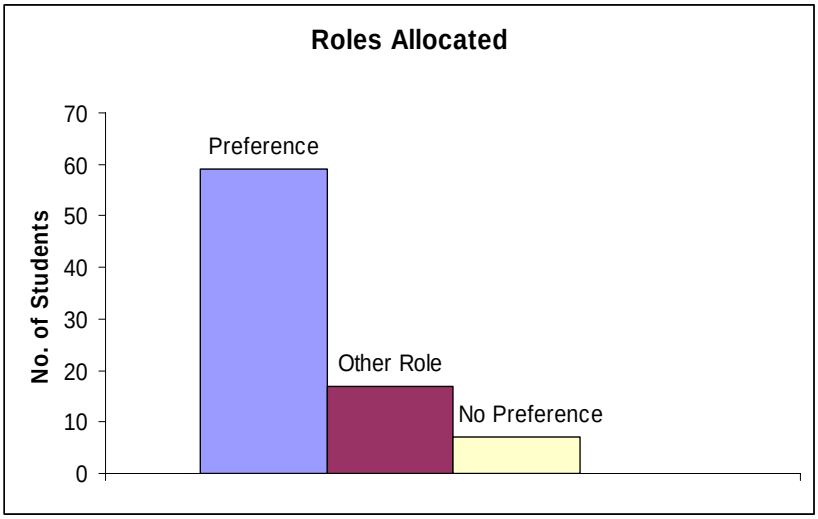


Figure 3: Roles allocated in area of preference at Interim Stage

At the interim stage of the project of the students who had expressed a preference, (71%) had been allocated a role within an area of their preference whereas 21% had been forced to take on a role outside that area of preference. The assigning of roles was left totally in the hands of the student teams and involved discussion of skills and preferences at their first onsite team meeting. Some students were allocated their roles by their team leader but allocation method really depended on the team structure and decision making policy agreed between teams.

During the first 5 weeks team structures in the Software Development Industry were identified and defined. These team structures were as follows:

Democratic Team - 'Egoless' team where all personnel are equal. Team reach consensus on all decisions

Democratic Team with Team leader - variation of Democratic Team where one person is designated as 'team leader' to coordinate the project. The team leader is

still regarded as a peer member of the team and consensus on all decisions is still required.

Autocratic Team - This team has one 'boss' in charge of the team who makes all the decisions and delegates tasks to specific team members.

Hierarchical Team - This team is 'in-between' autocratic and democratic in that there is a tiered hierarchy of seniority. The project leader assigns tasks and makes decisions but with the cooperation of other members of the team. This type of team is particularly well-suited to top-down development.

Chief Programmer Team - This team is usually part of a highly structured, hierarchical organisation. Usually consists of small team size and consists of a highly specialised and qualified team leader - normally chief programmer along with other skilled specialists.

Students were asked to indicate individually which team structure they thought would be most suitable for this particular software development project and indicate reasons for their choice. A majority of students chose the Democratic team structure with the variation of having a team leader. In discussion of their choices the main reasons given were they felt they were all peers and at the same point in their careers and therefore decision making power should be equal for all members. They also acknowledged that a team needed a leader or someone to coordinate activities if it was to be successful.

Team Structure	Individual Preference at Beginning (83)	Actual Team Structure - Team Interim Report (12)
Democratic	10	2
Autocratic	0	1
Hierarchical	19	6
Chief Programmer	3	0
Other	3	0
Democratic with team leader	48	3

Table 2: Team Structure Preference

At the interim stage of the project it is surprising to note that the majority of teams decided on the hierarchical structure. The most common form of hierarchical structure outlined in their reports was based on the team having a programming and a documentation sub-team as students felt these were the most important areas and the focus of many of the deliverables. The choice of team structure reported at the interim stage varies significantly from the initial choices made by individuals and this is because teams managed to meet up and discuss how they were going to tackle the work. Team reports indicate that the decisions were reached by recognising what work had to be done and what skills were available within the team.

As tutors we need to help students to become the more 'rounded' individuals that employers expect. We need to widen their horizons and make them more aware of the possibilities in terms of their career choices. Few of the students involved in the module 2005/2006 would have considered exactly what skills they possessed prior to undertaking the skills assessment or have identified which role would be most suitable for them or which stage of the software engineering process they would be best to work on.

The assessments described here are a way of helping students determine what type of role they would like upon graduation. Many Computing Students do not recognise the ‘soft skills’ that they possess and may rule out certain roles because of this. The project design for 2005/2006 ensured that students got to evaluate their skill weaknesses and strengths at the outset and this evaluation enabled them to make choices with regard to the roles they took in the project. Many students found the skill identification useful especially when it came to determining the stage of the software engineering process their skills would be most suited to or they would find most interesting to work on.

The module also provided the opportunity to learn new skills, take on new roles, strengthen weaker skills and learn how to work in a team.

The cross-site and video conferencing elements of the 2005/2006 module meant that students learned to communicate, plan, negotiate and compromise within their teams and across sites and it is these team-working skills that are important and could differentiate between graduate candidates in the competition for jobs.

During the project students were able experience new structures and ways of working and get a view of how large team dynamics can work. The module provided a realistic experience with regard to problem description and with regard to the realities of working in teams located at a distance.

The cross-site dimension to the module and the use of video conferencing were not without their problems but gave a more realistic flavour to the module and really challenged team-working skills. Students at both sites had to consider another team besides themselves, they had to look at the bigger picture and find ways to make sure that each side was able to meet their requirements and fulfil their aims.

The design meant that students had to program a solution to the problem but also communicate, schedule, plan, organise and negotiate with their onsite team and their Durham counterparts and behave, in effect, like a large software company with minimal guidance and instruction. They had to take responsibility for their learning and their learning outcomes. They liked the project as a whole especially deciding on roles and challenging themselves and said they felt more confident about their abilities and in some cases, that they enjoyed finding out that they were capable of taking on totally different roles than they had expected.

Conclusion & Further Work

Realistic group-work experiences at undergraduate level enable an *authentic* and engaging learning experience. In this type of learning, “materials and activities are framed around "real life" contexts in which they would be used. The underlying assumption of this approach is that material is meaningful to students and therefore, more motivating and deeply processed.” [1] The Software Engineering Team project described here aimed to make undergraduate team working more realistic and relevant to employers and to engage students and encourage them to develop team-working skills as well as their technical skills. Adding a cross-site dimension to the team project introduced new challenges and emulated what happens in large software companies whose teams are located at a distance. The cross-site working went relatively well but technical problems and scheduling and assessment differences made collaboration difficult.

Based on student feedback and tutor experience a number of necessary refinements to the assignments and future running of the experiment were identified. These are as follows:

1. We need to re-design the module so that Newcastle students have more motivation for collaboration in terms of marks and assessments. Although the need for commonality between software look and feel was outlined for students, the difference between site development schedules and the fact that exact proportions of marks were not allocated specifically for collaboration made the reasons for collaborating unclear to some students.

2. On the whole scheduling posed a large part of the problems encountered and therefore we need to align our timetables so that enough time is available for arranging meetings between teams. It is hoped that the opening of the Newcastle CETL video-conferencing suite will help to alleviate most of the problems encountered by Newcastle students the next time the project runs. Students will be able to manage the facility themselves and will not have to rely on a staff member being free to supervise meetings. The new facility will also be dedicated to the ALiC project and therefore students will not have to compete with other university projects for allocation of the room.

3. The video-conferencing technology itself was easy to use. The fact that inferior hardware was used at both sites meant that it was viewed as unreliable and gave the impression to students that it was not a stable technology. The hardware problems have largely been resolved in time for the next run of the project - Durham have a new camera and microphone and Newcastle have a new video server. Students at Newcastle will be trained how to use the new video-conferencing facilities and if difficulties arise there will be a procedure in place to ensure that problems are resolved quickly and fewer meetings are cancelled due to technical problems.

4. Students need to be trained how to conduct meetings and this is something that needs further discussion.

5. Further alignment between sites is needed with regard to timetabling, assessment methods, curriculum emphasis and the scheduling of deliverables and these will also be discussed before the experiment runs again.

References

1. http://www.nald.ca/adultlearningcourse/glossary.htm
2. From The Influence of Active Learning Experiences on the development of graduate capabilities - David ember and Doris Y.P.Leung, Journal of Studies in Higher Education, Vol. 30, No. 2 April 2005, p155 - 170
3. From: Effective Learning & Teaching in Higher Education, Module 1, Parts 1 & 2, “What is Active Learning?” by Pam Denicolo, Noel Entwistle and Dai Hounsell, CVCP Universities Staff Development and Training Unit, Sheffield, 1992
4. Module outline document - get proper reference
5. Taken from Learning in Teams, A Student Manual, Graham Gibbs, Oxford Centre for Staff Development, 1994, ISBN 1 873576 20 X, which is based on Management Teams, RM Belbin, Heinemann, 1981
6. http://www.belbin.com/belbin-team-roles.htm - 29th March 2006
7. http://news.bbc.co.uk/1/hi/education/4748438.stm - 29th March 2006

   http://www.agr.org.uk/ - Association of Graduate Recruiters - 29th March 2006