Proceedings of the Informing Science + Information Technology Education Conference

An Official Publication
of the Informing Science Institute
InformingScience.org

InformingScience.org/Publications

July 31 - August 5 2017, Ho Chi Minh City (Saigon), Vietnam

FAIR ASSESSMENT OF CONTRIBUTION AND PROCESS IN STUDENT TEAM PROJECTS

Marie Devlin *	Newcastle University, Newcastle upon Tyne, United Kingdom	marie.devlin@ncl.ac.uk
Lindsay F Marshall	Newcastle University, Newcastle upon Tyne, United Kingdom	lindsay.marshall@ncl.ac.uk
Chris Phillips	Newcastle University, Newcastle upon Tyne, United Kingdom	chris.phillips@ncl.ac.uk

* Corresponding author

ABSTRACT

This chapter outlines two different methods of peer assessment used in team projects at Newcastle University. One of them requires team agreement, whilst the other allows students to assess each other anonymously. Our results show that students are not convinced about the validity and fairness of either approach, and this illustrates that we need to do more work on teaching them how to assess fairly and to reassure them about the benefits of this form of assessment.

INTRODUCTION

This chapter outlines the assessment methods used in Team Projects at Newcastle University, School of Computing Science. These methods have been introduced to ensure fairness in group assessment and to ensure that team processes are assessed as well as team products. Team projects take place in two modules at level 2 of our BSc Hons Computing Science (CS) and BSc (Hons) Information Systems (IS) programmes. One module is the Software Engineering Team Project that is completed by both CS and IS students, and the other is a video project that is completed only by Information Systems students as part of an Information Management module. The chapter begins by providing the context and motivation for the assessment methods we use. We then introduce each module and outline the specific approach and context to learning and teaching we use for these and detail the assessment methods used, including those for peer and self-assessment. The chapter then provides detail of the impact of these assessment methods, using student feedback taken from focus groups and module questionnaires. Finally we provide an overview of our future work to address the issues raised by students and we outline the assessment challenges that remain.

This is a republication of the following:
Devlin, M, Marshall, L, & Phillips, C. (2016). Fair assessment of contribution and process in student team projects. In J. Whatley & C. Nerantzi (Eds.), *Teaching with team projects in higher education* (pp. 89-106). Santa Rosa, California: Informing Science Press.

Literature Review

The Benefit of Team Projects

Team work assignments have become common practice in most HEI undergraduate programmes for the sound pedagogical reason that university education is not just about developing what people know and understand in isolation, but about learning from, and with, others for the benefit of society. Team working helps students "shift away from simple academic achievement to much broader goals – preparing them for their working lives" (Lejk, Wyvil, & Farrow, 1996). Team working is also essential for students who wish to enter the profession of software engineering, as the modern industrial software engineering process "typically involves participation of software designers, programmers, end-users and domain experts and is essentially, a team-based activity" (Layzell, Pearl Brereton, & French, 2000). The educational benefits of students working cooperatively in groups are well- recognized. These include the following:

- Studying collaboratively has been shown to directly enhance learning;
- Employers value the teamwork and other generic skills that group work may help to develop;
- Group work enhances student understanding, students can learn from each other and benefit from activities that require them to articulate and test their knowledge;
- Group work provides an opportunity for students to clarify and refine their understanding of concepts through discussion and rehearsal with peers;
- Working with a group and for the benefit of the group also motivates some students;
- Group assessment helps some students develop a sense of responsibility (James, McInnis, & Devlin, 2002).

The benefits of teamwork projects that James et al. (2002) outline are in some senses generic as they could suffice as motivations for all teamwork in Higher Education, no matter what the discipline of the students that are involved. However, these benefits are particularly relevant to computing students, who tend to spend the majority of their degree programme being assessed on their individual task performance. We introduce teamwork because this is very much how students from our degree programmes will work once they enter the software development industry. It is rare for software engineers to work in isolation. They need to work with professionals from other disciplines and other domains of interest, e.g., to implement a system that monitors pipelines for oil and gas, software engineers need to work with engineers from these industries. Part of what our students must also learn when working with others is the ability to assess their own performance on all the projects they work on. The ability to assess and evaluate their own performance means that as practitioners they should be able to continually improve that performance over time and thus reflection will lead to increased experience and knowledge. As software engineers our students will also be assessed on their performance by others in their workplace, so they need to understand the appraisal process and how it can help to further develop their skills as practitioners.

However, being assessed as part of a team during their degree programme can make students uncomfortable, especially if the process contributes marks to the overall classification of their degree at the end of their studies. Students often feel that they are relying on the performance of others to ensure a good grade, and this is viewed as a negative consequence of any team working activity. It is therefore imperative that any assessment methods used should clearly illustrate to students how they were personally assessed for their contribution and also how any marks for collaborative work are divided amongst team members.

Assessing Individuals in Teams

To produce a mark at the end for each individual taking part in a team project module we need to measure the effectiveness of the team *process as a whole* as well as the *product(s)* delivered at the end of the learning activity and then determine the part played by each individual in each of these. Converting a student's contribution on a group task into a numeric grade "is a complicated and problematic task" (Lejk et al., 1996), and part of the reason for this is because team learning cannot be 'captured' or measured as easily or in the same ways as individual learning can. It is common to assess individuals on team projects in Higher Education in the UK by using peer assessment as one of the methods to capture a picture of an individual's effort. This is because students in each team have a first-hand and, therefore, more rounded perception of how the team have worked together on the project. The team gets to experience the reality of each team member's contribution to the team effort and the team process mainly because a lot of their work for the project takes place outside the classroom and away from the observation of the teacher.

As a method of evaluation, however, peer assessment can cause its own problems. Students often find it hard to be objective about their own performance and that of others, especially if the assessment given will impact on their own or teammates' grades. Freeman, Hutchinson, Treleaven, and Sykes (2006) argue that to ensure peer assessment works as a positive and effective experience for student teams, we need to address design and support issues that arise and thus allow students to improve their ability to make judgements on what constitutes good teamwork. Coppit (2006) outlines the assessment system for team work in software engineering that he feels should be used. The assessment must be fair and "quantitative and objective as possible, with little grader bias". Coppit maintains this goes for all assessment of teamwork, be it assessment by the tutor or by the students. However, it is difficult for students not to be biased when allotting quantitative values to themselves or others, as they are aware the values will impact directly on marks. For this reason, peer assessment is not the only form of allotting marks to students or on giving feedback on performance that should be used. Peer marks or grades should be only one method of evaluating a team or an individual's performance. Tutors need to ensure that other methods are in place, to get a more rounded picture of what is happening in teams. Coppit also advises that "the assessment scheme must not be easily manipulated by students and students should feel they receive the grade that they earn, good or bad." Again, it can be difficult for students to understand their grade if peer assessment is only numerical and is the only method of assessment used. Students can sometimes try and manipulate peer assessment by *collusion* with others, e.g., "if you give me a good grade, I will give you a good grade" So, it is important that other ways of determining individual performance are put in place, e.g., observation, individual reports or log books, and project viva with a member of staff. It must also be clear to students how their final marks are derived from the variety of assessment methods used. Any marking scheme for team projects should also enable the grading of different types of work, for example, team leadership, documentation, software implementation or testing, i.e., both the product of the project and process of the project. Any system that supports and facilitates the peer assessment process should help students understand the purposes of peer assessment and how the assessment criteria should be applied. The peer assessment process should ideally be viewed as another aspect of learning for students, rather than as just another mechanism that allows teachers to arrive at a summative mark for each student. Good systems should allow students to objectively review the outcome of a project, to determine where improvements could be made for the next project, to discuss what could be done differently to make a process more efficient, and to evaluate skills that team members may need to improve – and these are all aspects that need to be considered in professional project teams, no matter what the discipline. The ability to give constructive evaluations to others is a skill employers will value and students will be expected to use in the workplace, e.g., when reviewing their own annual performance or that of a project team they are leading. Peer assessment of team work also allows students to free themselves from depending on the authority of the teacher and thus develop intellectual independence and maturity through interaction with their peers. At Newcastle we use several methods to understand the team process more accurately and to get a clearer picture of each student's contribution in team projects. Team projects take place in two modules at level 2 of our BSc (Hons) Computing Science (CS) and BSc (Hons) Information Systems (IS) programmes. One module is the Software Engineering Team Project that is completed by both CS and IS students and the other is a video project that is completed only by IS students as part of the Information Management module.

The Software Engineering Team Project Module

According to Weinberg (1971), the worst way to run a programming project is "to hire hordes of trainees and put them to work under pressure and without supervision – although this is the most common practice today." Aspects of this statement still very much describe current undergraduate team projects in software engineering in Higher Education. Students who have never experienced teamwork or team software development are put together in inexperienced teams, given tight deadlines, and asked to develop a software system. The one redeeming thing perhaps, in contrast to Weinberg's view, is that we do tend to supervise student teams quite closely.

The Software Engineering Team Project at Newcastle is a year-long project where we randomly allocate students to teams and ask them to create and document a large software product. The module is worth 20 credits (10 ECTS) at Stage 2 (second year of our undergraduate degree programmes). Prerequisites for the module include a combination of programming modules covering Problem Solving, Program Design and Implementation, and Object Oriented Program Design Development. All students also complete a core module entitled The Software Engineering Professional, which includes topics such as ethics and professional and legal standards for software development. The motivation for the module is that the software engineering discipline has a strong team-based focus and the module needs to reflect industrial practice in a useful way in order to both enhance the students' learning experience and increase their employability. Students need to learn how to work with others to develop software collaboratively and this involves learning how to manage their time, make decisions about the planning and scheduling of activities, allocate work fairly based on strengths within the team, and find ways of working with other people's design ideas and code. The module also aims for students to learn the 'soft' skills associated with teamwork including negotiation, communication, delegation, leadership, and professionalism. The specific intended knowledge and skill outcomes for the module are the following:

- an understanding of the issues that relate to the planning and execution of a team-based software project.
- practical experience in issues such as team structure, document preparation, project management, and the design and implementation of a large software system,
- the ability to work as a member of a team, to fulfil appropriate roles and to apply these and other skills to the job at hand.

Team members take on relevant roles, such as project manager, software tester, or documentation lead, and together they allocate tasks, plan the project, and determine the development methodology. Examples of the type of project students have been asked to create in the past include mobile applications for language learning and a system for educating young people about health and fitness. We expect student teams to work almost independently, but they get support when needed from staff via weekly formal meetings for each team leader and from project *clinics* where all team members can come and talk to the module leaders about their project and any particular issue they are facing. Common problems faced by teams include perceptions of unequal effort between team members, lack of attendance or participation by some team members, poor task completion, or disagreements about workload allocation and sharing.

Evidencing Effort and Assessing Learning

For the first five weeks of the Software Engineering Team Project module, students are taught software engineering theory in lectures, and these provide a broad flavour of the discipline. Lectures include an overview of project planning and management, process models, and team structures as well as the stages of the software engineering process (Requirements Analysis and Specification, Analysis, Design, Implementation, Configuration Management, and Testing and Debugging). Students are placed in their teams in week 3 of the first semester, and, after the five week lecture series is completed, they are expected to meet in their teams at least once a week for an hour, in a 'formal' meeting to work on the project management activities (these include planning, task allocation, communication, and task completion). The module has quite a few assignments to keep teams busy throughout the academic year. Assignments are in the form of project 'deliverables' and in Table 1 these are mapped to the module learning outcomes to illustrate where these are assessed. Deliverables comprise individual submissions (I) and team submissions (T).

Table 1: Mapping of Learning Outcomes to Assessment of the Software Engineering Team Project module.

Learning Outcomes	Project Deliverables / Assessment methods
Communication – with customer, Problem Solving, Requirements Analysis	Requirements Analysis/ Domain Analysis (T)
Use of initiative, project planning, use of software development models	Project Plan (T) Log books (T, I), team reports (T)
Software Design, use of industry standards and practices for design notation	Specification document (T)
Programming in a team, testing, software development practice	Software source code and test documentation (T)
Adaptability, leadership, interpersonal communication, work in designated role as a team member, time management, personal organisation	Personal Skills Essay (I), Individual Reflective Report (I), Monitor observations, Peer and Self-assessment (I, T)
Presentation Skills	Team presentation (T), Trade Fair (T), Technical demonstration to customer (T)

Contribution Matrices

To accommodate students' concerns and make individual assessment fair, we specify a simple contribution matrix for inclusion with each team submission (Figure 1). This provides the opportunity for teams to describe individual members' contributions for every deliverable, i.e., who was responsible for creating, modifying, editing, or reviewing documents and code associated with the project. Teams can add other actions and tasks according to the nature of the assessment – e.g., technical tasks like programming sections of code.

	TA	BB	DG	FM	PM	NS	GS
Section 1: Planning	C, M, R	R	R	R	R	R	R
Section 2: Communication	M, R	R	R	R	R	R	C, R
Section 3: Task Allocation	M, R	R	R	R	R	C, R	R
Section 4: Meetings & Morale	M, R	R	C, R	R	R	R	R
Section 5: Problems and Successes	C, M, R	R	R	R	R	R	R

Key (C = Create, M = Modify, R = Review

Figure 1: A sample contribution matrix for a Team Report

The matrix clearly shows which parts of the deliverable each team member produced. It can also record team processes, e.g., organisation or leadership. Everyone in the team must agree to the matrix before it is submitted. We use the matrices at the end of the module to verify the overall contribution of each student to all team deliverables. They are generally used in conjunction with observation at formal meetings and presentations as well as team documentation, such as minutes of meetings, team log books and reports.

Contribution matrices are used only to adjust an individual's marks if there is disagreement at the end of the project as these verify the exact nature and extent of their efforts. Students can also use these matrices themselves to help decide on peer assessment percentages during or at the end of the project.

Allocating Percentages: Divide 100%

As the teams manage themselves outside of formal classes and project meetings, we need to ensure that we capture this informal learning and activity and that it contributes to the assessment of each student's performance at the end of the module. We use our peer assessment activity to do this. Students conduct peer assessment twice during the project – mid-way through the first semester (at the early stages of teamwork, where teams are getting to grips with the challenge of working together) and at the end of the project when students have delivered their software and the teamwork is almost complete. We ask students to divide 100% between team members (including themselves) based on their perception the effort and contribution to the project processes and deliverables each member has made. Figure 2 shows an example percentage allocation for a 5 person team. In this example, the column on the right with the heading "total/5 to 1 decimal place" adds up to more than 100% but this is deemed insignificant: it is hard to be precise, but we round up above 0.5. The marks for all team assignments are multiplied by these peer marks, as a team weighting, to get the student's overall individual mark for the team deliverables, e.g., if a design document received a group mark of 77%, Joe would get the full mark for the assignment, given that he has the highest peer weighting. The module leader also assesses effectiveness during the project, as a team, based on performance in meetings and presentations throughout the year and on each team deliverable. The effectiveness assessment is considered together with the peer percentage weightings to decide each student's overall mark for team deliverables. Individual assignments for the module are not subject to weightings or peer review.

Peer Percentages Allocated **to**	Peer Percentages Allocated **by each student – Anonymously**					total/5 to 1 decimal place
	Student A	Student B	Student C	Student D	Student E	
Joe	24	20	25	16	28	22.6
Jack	15	20	17	20	20	18.4
Jane	16	20	17	19	16	17.6
Jean	20	20	26	16	15	19.4
Jim	25	20	25	29	21	24
	100	100	100	100	100	

Figure 2: An example of percentage allocation

The Information Management Module Team Video Project

The second project in which students peer assess is a video making project that students undertake in the Information Management module. They must create a 5 minute video loosely relating to a specified theme. The aim of the project is to experience the technical aspect of film making and also the legal issues that arise, e.g., music copyright and filming in public places, as well as the important aspect of teamwork. We place students into teams of 5-6 students and ask them to produce several assignments relating to their final video submission.

The students receive an introduction to storyboarding, script preparation, and the organisational issues that they need to think about when creating a video, such as continuity and timing. There is no specific teaching with respect to the technical issues of video production such as using cameras and video editing – students frequently use their own equipment and software so it would not be especially useful to them as there is considerable variation in features and interfaces.

There are four pieces of assessed coursework: an individually prepared storyboard, a script created by the group based around team members' storyboards, the video itself, and an essay reflecting on the student's own experiences when working on the project. The peer assessment values are applied to the mark that the team receives for their video.

Allocating Percentages: "Show Me the Money!"

In order to review their team process via peer assessment we ask students to distribute an imaginary £1000 bonus between team members according to their perception of each person's effort and contribution to the team goals, each student anonymously allocating a proportion of the £1000 to their team mates, excluding themselves. The tutor then finds the maximum amount allocated to one student and that student gets the full mark given for the project, e.g., 16 out of 20. Everyone else gets the marks times their total divided by the maximum amount. So, for example, if a maximum mark is 16/20 for a piece of team coursework the allocations for a team of 6 students could be as shown in Figure 3. In this example student E would get the full 16/20, student A would get 15/20, and so on.

	A gives to	B gives to	C gives to	D gives to	E gives to	F gives to	Total for student	Total as propor portion of highest	Proportion converted to grade
A	----	£200	£237.50	£275	£180	£200	1092.5	0.94	15.0
B	£200	----	£237.50	£225	£240	£200	1102.5	0.95	15.2
C	£250	£200	----	£200	£240	£200	1090	0.94	15.0
D	£250	£200	£237.50	----	£240	£200	1127.5	0.97	15.5
E	£200	£300	£237.50	£225	----	£200	1162.5	1.00	16.0
F	£100	£100	£50	£75	£100	----	425	0.37	5.8

Figure 3: Allocation of money in Information Handling

METHODS FOR GATHERING FEEDBACK

To determine the student experience of peer assessment methods used during the team projects at Newcastle we reviewed module questionnaires, conducted focus groups, and reviewed individual reflective reports for all students taking part.

- **Module Questionnaires** - All students who participate in these modules complete an online evaluation of the module anonymously. Team project modules are surveyed twice during the academic year, once before Christmas and again towards the end of the final term, before the exam period. These questionnaires contain some standard questions on the module, including questions on the quality of lecture delivery, structure of modules, etc. It is in the free-text areas of the questionnaires that students tend to comment on their feedback and assessment experiences, and we focused on both these areas when gathering responses.
- **Focus Groups** - The focus group was conducted using 1 participant from each team for the Software Engineering Team project (i.e., for 22 teams we had 22 participants and ran 2 sessions of the focus group). The focus groups were facilitated by a ‘neutral’ facilitator. We have run several of these focus groups over the years and normally ask an employer to host these for us. Students are comfortable talking to someone who has acted as a ‘customer’ during the year but who has no impact or influence on their assessment.
- **Individual Reports** - Each student on these modules must submit a report at the end of the year (approx. 4 pages in length) reflecting on their individual experiences during each module and also on their performance in their allocated role.

RESULTS

From the questionnaires and focus groups we found that students felt one of the best things about the team work modules were that they were able to work with and get to know their fellow students. They also liked the project briefs and thought they were interesting and challenging, and they also thought they prepared them well for working in real business situations. They also felt that the skills gained from the modules were good for their CVs and helped when interviewing for internship or placement positions.

Some students thought that the idea of randomly assigning students to their teams was a good idea and that, even though the work was sometimes stressful, it was fulfilling. Students commented that the modules were “completely unlike all the other modules because they teach you about the personal and organisational areas of projects, instead of just the technical” (Survey comment). Team members commented that they were pleased to be able to design a product with their own designs rather than just being told what to do. They also felt the modules pushed them to behave differently, as one student observed when asked to specify something good about the modules (Figure 4):

> *"Being put outside your comfort zone. You force yourself to learn new things, improve and develop skills. "*

Figure 4: Student feedback on module questionnaire

The results showed that the first peer assessment, which takes place early on in the software engineering project, acted as a motivator for some students, e.g., in the feedback in Figure 5, students were motivated to improve their performance after receiving a poor evaluation from their teammates.

> *"I felt the first percentage given to me (twelve) at the end of semester one by the other members of the team was justified. I felt this mark was low but it did drive me to contribute more in semester two and I had a genuine enthusiasm to obtain a higher percentage in the next peer assessment. I put in a great deal more effort into the project during semester two, participating more with group deliverables and contributed hugely with the team website and as a result I obtained a higher percentage in the second set of company percentages."* (Student 1)
>
> *"What I learned from this is that in a teamwork scenario, if there is any kind of peer assessment process, it should be the LAST thing to do. I think it worked well having a set of percentages both at the start of the year (end of 1st term) and at the end of the year. This mean that those that hadn't quite pulled their weight got this demonstrated through the initial percentage allocations and put the extra effort in in the second semester."* (Student 2)

Figure 5: Student feedback on their first semester peer assessment experience

When referring to working with others though, some students felt that there were areas of the projects that needed to change. Students commented that they would like staff to punish students that did not pull their weight, or that peer assessment should have a greater influence on marks. They felt that there needed to be some way of observing students outside of formal meetings, when members of staff were not present. Student feedback also showed us that the peer assessment process often had a demoralizing effect on the team with suspicion of collusion between members to get a better mark. These suspicions were not optimal for the continuation of work during the second semester, and the process sometimes caused tension among team members who then had to continue working together (Figure 6).

> *"I was not happy with my original peer percentage for the second semester which was 14.4 mostly because I felt like I had done more work than some others…Eventually we all (reluctantly) agreed on a mark that was fair. Personally I don't think this is a good system as it can be abused easily, for example two members could unfairly rate each other."* (Student 3)
>
> *"Peer review forms seem harsh and unhelpful in the team environment, even done anonymously it can cause rifts. If we must mark each other with percentages it should be done individually rather than as a piece of team work."* (Student 4)

Figure 6: Student Feedback on the Peer Assessment activities

In contrast to the peer assessment method of percentage sharing used in the Software Engineering Team Project, there were little or no disagreements about the money-sharing exercise used during the Team Video Project in the Information Handling module. This may be because the exercise is anonymous, and students submit their individual views on contribution to our in-house virtual learning environment NESS (Marshall, 2003). The peer assessment exercise takes place at the end of the Information Handling module, after all coursework has finished and meetings have taken place. There was, however, some evidence that students had colluded before the process and module leaders received complaints to this effect. However, it was very hard to prove that collusion had taken place, given only anecdotal evidence. The fact that the students do not mark themselves using the "Show me the Money!" method was also interesting, because they do not evaluate their own performance, only that of others. Using this method, if all students are given equal marks, then that is fine. However, if the marks allotted to team members differ and one attains a much higher mark than the rest, then there is a danger the assigner's mark could be detrimentally affected. So, there are areas of concern with this method that also still need to be addressed.

Evaluation

The peer assessment methods we used had a mixed reception from students. The peer assessment exercises could be positive and motivate or warn students who were not pulling their weight. However, they could also negatively impact team productivity and morale. One of the problems with our methods is that we did not give our students much guidance on peer assessment. We did give them tips about making fair decisions, e.g., we asked students to focus more on tasks, attendance, and performance rather than personality and behaviour. However, we really asked them to judge a person's performance without providing suitable and clear criteria for doing so. Students were, therefore, unsure of the reasons for peer assessment and generally found it difficult.

One way to increase students' engagement with the peer assessment process could be to illustrate the real value placed on evaluating performance as a skill needed in the real world (e.g., in software engineering or information handling roles). We found it interesting that student feedback from the Video Project peer assessment exercise was mainly positive, in contrast to feedback on methods used in the Software Engineering Team Project. We are, therefore, interested in evaluating if the reasons for the positive feedback from "Show me the Money!" are (a) because students used money instead of percentages or (b) mainly because the process was anonymous and online and, thus, avoided any confrontation or face to face discussion and the need for agreement between team members.

The use of peer assessment during each of the project modules was, in part, to help us estimate an individual's contribution for summative purpose, but mainly to help students reflect on their performance and what they achieved and learned. However, our students' experiences of peer assessment were somewhat negative and counterproductive in some cases. We need to improve their experience by retaining the good aspects of peer assessment as a learning mechanism and by working on the weaker areas of the exercises. It may be that our peer assessment methods (especially that for allocation of percentages) are not working as well as we hoped in combating collusion or in helping students learn evaluation skills.

There are weaknesses in both approaches in terms of engaging students with the process and in our reassuring them about fairness. Also, despite our efforts to ensure that both products and processes are assessed in these projects, we still face difficulty in capturing all effort and student learning throughout each project. Whilst students generally reflect on their experiences as useful and productive in terms of learning new skills, they still tend to experience difficulty in expressing to what extent their skills have developed and changed throughout these projects.

Conclusion & Future Work

We have outlined the peer assessment methods we currently use during Team Project modules in Computing Science at Newcastle and have given some examples of student feedback on their experience using these methods. The anonymous submission method used in the Information Handling module was the most successful in terms of helping students feel comfortable and reassured about the peer assessment process. The need for team agreement on contribution matrices in the software engineering module was also successful in terms of recognition of individual effort. It is these two elements that we think the reader might want to try out in their own team projects.

We hope to conduct a study that will allow us to further explore different peer assessment mechanisms to ensure they are a positive learning experience for students and this includes new ways of assessing and evaluating skill development during these types of modules. We are currently working on our own online system that will allow students to evaluate each other anonymously and to select one area of strength that they personally would like to be evaluated on. This new system should remove the focus on percentages and numbers and allow students to focus on assessing their own and their teammates' competency-levels for a variety of skills and abilities that are related to the module learning outcomes. We also will continue work on ways to prevent collusion during peer assessment.

References

Coppit, D. (2006). Implementing large projects in software engineering courses. *Computer Science Education, 16*(1), 53-73.

Freeman, M., Hutchinson, D., Treleaven, L., & Sykes, C. (2006). Iterative learning: Self and peer assessment of group work. *Proceedings of 23rd Annual Conference Who's Learning, Whose Technology?* 2006, pp257-266.

James, R., McInnis, C. & Devlin, M. (2002). *Assessing learning in Australian universities: Core principles of effective assessment*. Retrieved 08/07/2013 from http://www.cshe.unimelb.edu.au/resources/assessment

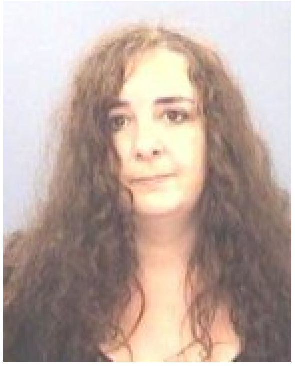
Layzell, P., Pearl Brereton, O., & French, A. (2000). Supporting collaboration in distributed software engineering teams. *Proceedings of Seventh Asia-Pacific Software Engineering Conference, ASPEC 2000*, pp 38-45.

Lejk, M., Wyvil, M., & Farrow, S. (1996). Survey of methods deriving individual grades from group assessments. *Journal of Assessment & Evaluation in Higher Education, 21*(3), 267-280.

Marshall, L. F. (2003). NESS - The Newcastle E-learning Support System. In the *4th Annual Conference of the LTSN Centre for Information and Computer Sciences*. NUI Galway, Ireland: LTSN-ICS.

Weinberg, G. M. (1971). *The psychology of computer programming*. New York: Van Nostrand Reinhold Company.

Biographies



Marie Devlin has worked in Higher Education for 17 years. She joined the School of Computing Science at Newcastle in 2005 as a Teaching Fellow in the Centre for Excellence in Teaching and Learning Project: Active Learning in Computing and has been researching approaches to teaching team projects since then. She was promoted to Lecturer in 2012 and is an Academic Associate of the HEA and a Senior Fellow.

Lindsay Marshall joined the Computing Science Department at the University of Newcastle upon Tyne as a Research Associate in 1980 and whilst working as an RA was responsible for the majority of the development of the Newcastle Connection software. He was promoted to Senior Lecturer in August 2003 and was awarded the Vice Chancellor's Distinguished Teaching award in 2014. He is a Senior Fellow of the Higher Education Academy.

Chris Phillips is Professor of Computer Science Education at Newcastle University. He joined the Department of Computing Science in 1984 as Lecturer, then Senior Lecturer, and became Dean of Undergraduate Studies in the Faculty of Science, Agriculture and Engineering in 2009. Since the start of 2014 Chris has been on secondment as Head of Academic Operations at Newcastle University International Singapore. Chris is a Senior Fellow of the Higher Education Academy and a Chartered IT Professional.