Title:
EXPERIENCES OF INTRODUCING GROUP PROJECTS TO COMPUTING DEGREES

Theme:
Undergraduate group project work.

Authors:

*Martin Shepperd and Amkbikesh Jayal
Dept. of Information Systems & Computing
Brunel University
West London, UB8 3PH
UK
{martin.shepperd, ambikesh.jayal}@brunel.ac.uk

* Author who will be the primary contact
EXPERIENCES OF INTRODUCING GROUP PROJECTS TO COMPUTING DEGREES
Martin Shepperd and Amkbikesh Jayal
Dept. of Information Systems & Computing
Brunel University
West London, UB8 3PH
UK
{martin.shepperd, ambikesh.jayal}@brunel.ac.uk

ABSTRACT
As a result of restructuring our computing undergraduate programmes, we introduced group projects as an integrating central spine for Levels 1 and 2. The task includes requirements discovery, software development, team working, negotiation and project management. We found careful team formation and fostering of close staff-group relationships to be invaluable. The outcome has been an improved sense of engagement for both students and staff plus improved student technical skills arising from the experiential learning.

1. INTRODUCTION
This paper is not intended as a scientifically rigorous study but rather an informal account of our experiences that we hope may (i) have some motivational value and (ii) offer some useful ideas and techniques for others involved in group project work in universities. We are located in a computing department which brings particular challenges and opportunities, however, believe many of the principles are relevant to other disciplines such as engineering, design, business studies, psychology and architecture.

These ideas and experiences derive from a thorough redevelopment of the undergraduate curriculum for two undergraduate computing degree programmes (BSc Computer Science and BSc Information Systems) during 2008-2010 at Brunel University in London. This present academic year (2009-2010) represents our first experience of the new scheme. At its heart lies a substantially increased level of project work (approximately a third of each year) acting as a spine to the entire programme.

Clearly the idea of students working as a group to carry out a software project is not new. A pioneering paper (Freeman, Wasserman, and Fairley 1976) highlighted the need for computing students not only have strong technical skills but also problem solving and communication skills. They also pointed out the substantial differences between an academic and industrial environment for software developers. For these reasons they argued strongly for the introduction of group projects into the computing syllabus. A particular innovation that they promoted is the software hut (the name suggests a small software house) where students are expected to 'sell' their software or solution to others and that some proportion of the marks are derived from this exercise. The idea is to introduce elements of marketing and competition.

Other more recent work that has championed the use of undergraduate group projects is to be found at the University of Sheffield in the UK (Holcombe, Gheorghe, and Macias 2003). Here they take the idea of the software hut to its logical conclusion and actually have student groups working for external clients (though of course no
money is involved). A detailed protocol is agreed but essentially the client gets to choose the best software from 4 or 5 teams and keep it. Of course this requires a good deal of maturity from the students so such projects are placed in the final year of their undergraduate programmes.

2. BACKGROUND
In 2008 the computing department at Brunel University decided upon a root and branch overhaul of their undergraduate programmes. This was prompted by:

- The limitations and frustrations of teaching large classes where even seminar groups can exceed 20 students
- Inconvenient timetabling due to lack of large lecture theatres
- ‘Uninteresting’ or dated modules¹ not reflecting current software development practices
- Lack of opportunities for students to display originality and creativity
- Students poorly prepared for their individual final year project
- Lack of connectedness or a sense of community so for example, even coursework is submitted and feedback received electronically via the university virtual learning environment (VLE)

This last factor was seen as especially serious, leading to demotivation and a lack of engagement both on the part of the student body and also, dare I say it, on the part of the academic staff employed by an explicitly "research intensive" university.

In addition we had to consider what sort of student we wished to graduate. This was addressed from two perspectives. The first perspective considers student long-term aspirations and the role of the degree in supporting their career. The second comes from our perspective and reflects what we consider to be core components of computing as a discipline.

The reality is that the majority of our graduates see themselves as heading for a commercial career rather than an academic one. This should not, however, be interpreted as a ‘dumbing’ down of the degrees to meet the needs of immediate employability. First jobs are only a stepping-stone to where our graduates belong – within 5 to 10 years of graduation they should have risen to leadership positions where they will carry significant responsibilities. This relates to SFIA² Level 5. These levels indicate degrees of autonomy and influence exercised by such staff within the organisation and in the draft version 4 of the framework they are defined as:

**Autonomy:** Works under broad direction. Full accountability for own technical work or project / supervisory responsibilities. Receives assignments in the form of objectives. Establishes own milestones, team objectives and delegates assignments. Work is often self-initiated.

1 A module is a named unit of content e.g. Introductory Programming and typically 5 or 6 modules combine to make a level (or year) comprising 120 credits.

2 The Skills Framework for the Information Age (SFIA) provides a common reference model for the identification of the skills needed to develop effective Information Systems (IS) making use of Information & Communications Technology (ICT). See URL: http://www.sfia.org.uk/
Influence: Influences organisation, customers, suppliers and peers within industry on contribution of specialisation. Significant responsibility for the work of others and for the allocation of resources. Decisions impact on success of assigned projects i.e. results, deadlines and budget. Develops business relationships with customers.

Achieving these levels of autonomy and influence depends not just on the degree but also gaining experience and maturity in more junior roles. Perhaps the most telling element of the SFIA level 5 definition in terms of the preparation that comes from our degree is its description of the ability to handle complexity:

Complexity: Challenging range and variety of complex technical or professional work activities. Work requires application of fundamental principles in a wide and often unpredictable range of contexts. Understands relationship between specialism and wider customer / organisational requirements.

From a discipline perspective, the core that anchors and distinguishes computing (or at least the department’s particular viewpoint):

A focus on software artefacts that exploit the capabilities of networked computers for the benefit of some human activity or purpose.

This leads to two families of degrees: those that are driven by a primary interest in the creation of software artefacts (computer science) and those driven by a primary interest in their beneficial impact on human activity (information systems). Each includes the other perspective as an essential secondary element anchoring them in the ethos of the Department and distinguishing them from the related disciplines.

The Group Projects provide a spine for Levels 1 and 2. We have groups of 3-5\(^3\) students and involve a task that integrates much of the other material in the year. The group project comprises 40 out of a total of 120 credits for the Level and thereby gives the student the opportunity to engage with a task of significant complexity. Typically this will include software development, context, communication, team working, project management and the application of different research methods (e.g. market research, usability assessment, etc.). Each team is supervised by a member of academic staff with regular contact (normally weekly), thus, we see the projects as the vehicle for traditional style tutorials.

This group project seeks to integrate skills covered by the Level 1 modules into a non-trivial, practical group task including a significant degree of programming and technical engagement. It is also intended to help the student gain confidence in their technical abilities.

3. MECHANICS
In this section we now describe in more detail the operation of the Level 1 group project. The students were informed that the aim of this 40 credit module (i.e. one third of the first year) was to:

\(^3\) We avoided larger groups since this made it easier to ensure students could not evade tasks about which they did not feel confident e.g. programming. In addition independent studies (Shaw and Harkey, 1976) have reported small team sizes of 3 to 6 are more satisfactory than larger groups (better bonding, less competition).
integrate skills covered by the other Level 1 modules into a non-trivial, practical group task including a significant degree of programming and technical engagement. It is also intended to help the student gain confidence in their technical abilities.

The learning outcomes for the module are as follows. Students should be able to:
1. plan, manage and track a non-trivial group activity.
2. take an open-ended problem and define and refine the requirements.
3. design, develop and test a modest piece of software (order of a few hundred LOC).
4. independently (i.e. individually) understand, modify and test a given piece of software.
5. effectively present, communicate and market ideas and solutions to their peers.
6. create and use technical documentation.
7. understand and apply the principles of professional and ethical behaviour in a group context.
8. reflect and learn from their group project experiences.

A number of aspects of the Group Project are intentionally open-ended to encourage student creativity and develop open-ended problem solving skills. Typically the coursework includes some requirements analysis and understanding of the problem context, software design and development, communication, team working, project management.

This module is intended to be primarily practical in its nature; in other words students learn by doing and, most importantly, reflect after-the-event. We try to reassure students that on occasions they will be asked to do something new and that this may seem daunting. However, it is by actually engaging with a problem that they have the best opportunity to learn and it is interesting to note that many of them commented favourably upon this aspect of the module. We appreciate the challenges students face and so don’t expect perfection.

Level 1 students were allocated to groups of four with the goal of creating teams of mixed ability and experience. Groups were then provided with a description of a problem such as a project management and planning tool. Teams then spent half the semester researching into the problem domain and developing a class library to underpin any detailed development. An important book that we recommended to the students to provide guidance on surviving as a team is Levin (Levin 2005).

Teams then endeavoured to ‘sell’ their part solution to other teams and to ‘purchase’ software from another team upon which subsequent development was required to be based. Marks for the first stage were related to the number of sales. Students quickly appreciated that documentation and a good design gave them a competitive advantage. The more innovative offered additional blandishments such as support contracts.

In addition, an outside software house visited the “Software Fair” and awarded prizes to some of the best teams. Next more detailed requirements were provided and the second part of the development based on the ‘bought’ code. The marking scheme was
weighted such that students could recover from zero sales from the first stage but be challenged if they made an unwise ‘purchase’.

Thus the students had an opportunity to learn about team working, planning and monitoring, requirements, marketing, procurement, documentation, testing, maintenance and changing requirements in addition to the chance to develop a non-trivial piece of software.

3.1 Team Formation

Early on in the academic year, the students were placed in a team of approximately four students and assigned a Tutor (who is an academic member of staff). Initially we formed as many 4 person teams as possible, however where boundary problems occurred we created 5 person teams. A couple of students left or transferred course very early on resulting in two 3 person teams.

The teams were chosen by us. Our intention was to form teams of mixed experiences and abilities, e.g. we mixed BSc Computer Science and Information Systems students. The reasons for this were twofold. First, it provides more opportunities for students to learn from their peers and second, it better represents ‘real’ software development teams. A particular problem for a Level 1 group project is that the abilities of students may not be known in advance. We handled this by delaying team formation until Week 4 and also by asking the students to complete an on-line diagnostic test. A few students failed to do the test and this (rather than the actual score) turned out to be a very good predictor of future problems.

We also informed the students that a special pool of students would be formed from non-attenders and non-participants (without valid reason) and used to form teams. This meant that if they were lazy and expected to be carried by their fellow students then they would be placed in a team with like-minded students. Our advice was therefore to engage in this important module right from the start. This was probably the single most important decision we made and turned out to be fully vindicated. Students who commenced with a laissez faire approach did not substantially change their behaviour⁴. It also meant that to a large extent we were able to ring fence problems into a very small number of groups rather than experience widespread impact due to the difficulties of a few students.

As far as possible team membership was stable throughout the year but if circumstances dictated (e.g. serious illness) we retained the option to make modifications. In this eventuality students could be compensated by the marking process. In practice we moved students between 4 teams (out of a total of 47) during Week 10. This was due to low levels of participation and external problems for six students.

To reiterate careful team formation was one of the most important decisions for this module and therefore we tried to use as many sources of intelligence as possible.

3.2 Group Project Tasks

⁴ We should stress such students were given, equal if not more support than others.
Next we describe the coursework the students were expected to undertake during this module. As far as possible we took an incremental approach rather than giving students a series of seemingly random and disconnected activities.

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
<th>Group / Individual</th>
<th>% of Total Marks</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1</td>
<td>Initial software prototype</td>
<td>Group</td>
<td>10%</td>
</tr>
<tr>
<td>T2</td>
<td>Integration of components and documentation</td>
<td>Group</td>
<td>10%</td>
</tr>
<tr>
<td>T3</td>
<td>Implementation of additional functionality and documentation</td>
<td>Group</td>
<td>50%</td>
</tr>
<tr>
<td>T4</td>
<td>Individual implementation of an unseen change to group system</td>
<td>Individual</td>
<td>Pass / fail</td>
</tr>
<tr>
<td>T5</td>
<td>Post-mortem style group review</td>
<td>Group</td>
<td>30%</td>
</tr>
<tr>
<td>T6</td>
<td>Assessment of ethical and professional behaviour</td>
<td>Individual</td>
<td>Pass / fail</td>
</tr>
</tbody>
</table>

Table 1: Individual Group Project Task Details

A difficulty we had was choosing something that was challenging for those who already had a strong background in programming but not impossible for those without prior knowledge. In the end we decided upon the development of a website suitable for new students at Brunel University. This commenced with some simple static html and css files (Task 1), integration of other websites into a single portal (Task 2) and the addition of dynamically generated webpages using java and jsp (Task 3). More details may be found in Table 1 and the Appendix. These tasks are representative of much modern software development and use a mix of typical technology. This was emphasised to the students as a motivator when some struggled with Task 3.

Note the intentional increase of mark weightings so that teams taking a while to "bed down" were not doomed to certain failure and retained the possibility of a good score. Note also the significant weighting (30%) attached to the post project review. Here the intention was to give substantial credit for the ability to learn from failure as well as from success.

We also decided to implement a version of the software hut (Freeman, Wasserman, and Fairley 1976) described in the Introduction as part of Task 1. The idea was to encourage an element of competition, to allow a degree of creativity since the task was relatively unconstrained and finally to help students view software both from the perspective of a developer and from that of a consumer. Each team developed a website, all of which we then hosted and each team could choose three others to incorporate into their Task 2 portal. Choices might be influenced by the quality of the content, implementation, etc. The teams were then ranked according to the number of times they were chosen and marks for Task 1 awarded according (see Figure 1).
Task 4 was designed to prevent weaker students hiding within a team and not developing their own technical skills. They were asked to make two small, unseen modifications to the system under exam conditions. The task was simply marked on a pass fail basis and was not intended to be overly difficult, rather to set some minimum threshold.

Post project reflection is widely advocated (if not always effectively conducted) as a part of good software development practice (Dingsøyr 2005). In addition reflection is stressed as an extremely important part of experiential learning (Boyd and Fales 1983) so we incorporated a significant opportunity for groups to explicitly reflect upon their experiences.

Finally, the Task 6 assessment of professional and ethical behaviour was intended as a sanction in the event that a student completely undermined their team through lack of commitment or other unacceptable behaviour.

4. FINDINGS AND DISCUSSION
To give some idea of scale and the running of the module Table 2 gives some basic logistics. As can be seen a significant number of staff were associated and this meant a substantial amount of effort was required to brief everyone and deal with ad hoc queries. With hindsight it would have been useful to prepare an information pack for each member of staff.

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Students</td>
<td>191</td>
</tr>
<tr>
<td>Teams</td>
<td>47</td>
</tr>
<tr>
<td>Tutors</td>
<td>28</td>
</tr>
<tr>
<td>Graduate teaching assistants (GTAs)</td>
<td>10</td>
</tr>
<tr>
<td>No. of teams making all submissions</td>
<td>45</td>
</tr>
<tr>
<td>Weekly formal schedule</td>
<td>1 hr lecture, 1 hr tutorial, 3 hr lab session</td>
</tr>
</tbody>
</table>

Table 2: Group Project Logistics

We now consider the outcome of running the Group Project in more detail, particularly the themes of engagement and performance. We then review the results of a student feedback survey.

4.1 Engagement and performance
Here we consider first student and then staff engagement.
<table>
<thead>
<tr>
<th>Task</th>
<th>No. of teams submitting on time</th>
<th>No. of late Submissions</th>
<th>No. of missed submissions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task 1: Team website</td>
<td>47</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Task 2: Integrated website</td>
<td>47</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Task 3: Dynamic website</td>
<td>45</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Task 5: Team review and presentation</td>
<td>45</td>
<td>0</td>
<td>2</td>
</tr>
</tbody>
</table>

### Table 3: Coursework submission rates

As Table 3 indicates the module has a remarkable level of participation in terms of submitting coursework on time. Recall that this is a Level 1 module and typically a number (10% or more) of students struggle, drop out or change course without informing us. It is possible that the sense of not letting down one's team mates and the close relationship with a tutor has led to this much increased level of engagement.

![Figure 1: Distribution of Choices for Task 1](image)

### Figure 1: Distribution of Choices for Task 1

Staff engagement is more complex to comment upon, since there is no assessment or formal feedback mechanism. However an immediate benefit is that all staff know at least 4 or 8 undergraduate students personally. There were also many examples of staff providing a role beyond that of project tutor e.g. giving advice on future choices of modules, etc.

Some problems inevitably arose. The tight integration with other modules did not always go exactly to plan, e.g. there were some synchronisation issues with the Introductory Programming module. This might be avoided in the future by more careful planning with all lecturers associated with Level 1 prior to the start of the
academic year. We also had group tutors with varying technical expertise due to the fact that the department embraces both computer science and information systems experts. This led to some apprehension, however, we encouraged students to use the lab sessions as their primary source of technical input. These were supported by our team of GTAs. A side effect of acting as tutor was learning by osmosis and it does seem, at least anecdotally, that some staff have gained in expertise and confidence over the duration of this project.

4.2 Student Feedback
It is customary to solicit student feedback for every module at Brunel University by means of a standard student feedback form that is distributed towards the end of the academic year. Many of the questions elicit a quantitative response in the form of a Likert scale, e.g. how would you classify the standard of your lectures 1 … 5? Since they are not specific to group projects and need to be interpreted relative to other modules we focus instead on two open ended questions:

- what do you like most about the module?
- what do you like least about the module?

We obtained 44 responses out of 190 students (though 5 or 6 have dropped out / changed course during the year) so a 23% response rate. This low rate was largely the consequence of the module finishing in advance of the rest of the course and therefore students being more difficult to locate. Whether this resulted in some kind of bias is hard to say, however, we take the

**What do you like most about the module?**
29 out of 44 made some response and 15 did not complete this question (10 of the 15 did not wrote anything negative either). Since the responses were open we used an informal card sort technique to group related comments.

(i) Working in a team (15 responses)

"Good to experience working in a team"

One respondent illustrated the dilemma of team work by stating "Working in a team" as the best thing but also "The group members" as the thing he or she liked least.

"I was very lucky and had a great team"

The interesting thing here is that the student saw this as entirely externally arbited and therefore nothing to do with his or her contributions to team building. This contrasts with another respondent:

"working in a team to be able to develop this skill"

"I really enjoyed working in a team …"

It's easy to lose sight of the idea that education can be fun.

" … because I think I learn a lot more rather than working individually."
"Exploring coding in more details (sic) with a group – get help and learn at same time"

and

"Working in the team, share different tasks."

where the student explicitly mentions the positive aspects of sharing.

(ii) Task oriented aspects of the module (13 responses)
A total of 13 students (i.e. about 30%) mentioned some technical aspect or experiential learning of technical skills.

"Designing a website from scratch using different programming methods"

was a common response. Also:

"Practical programming. Planning. Deadlines"

Here we see an appreciation of the experiential aspects of learning (this was a common theme) but also of the value of a project management and self-management skills.

Despite our efforts to achieve a high degree of integration with other modules this was either not recognised or not seen as important, so the following was quite unusual:

"This module has motivated me to cope with all the other modules."

(iii) Other comments (1 response)
Finally, and perhaps surprisingly for a group project module, one respondent liked the lectures most of all!

What do you like least about the module?
As previously indicated we had 10 blank responses both for positive and negative features. There were 9 with a positive responses without a negative (c.f. 5 with a negative response but no positive). Again we grouped the types of response.

(i) Disliked aspects of their team (8 students)
These respondents expressed some variant of the other team members didn't contribute appropriately.

(ii) Disliked aspects of the task  (15 students)
The fact that more students had strong negative feelings about aspects of the task, than had about being placed arbitrarily in a team we found slightly surprising. Most remarks seem to relate to Task 3 (the overall task was split into a series of sub-tasks each with their own delivery date). Task 3 represented a substantial step up in difficulty.
"Task 3 was a big step up and was quite a struggle."

Some commented upon specific activities they didn't enjoy such as "Creating the documentation" or specific technologies such as the webserver Tomcat. It is likely the latter stem from frustrating experiences rather than deep problems.

Two students commented specifically commented on what they saw as excessive task difficulty, e.g.

"Quite a lot of expectations of understanding everything."

Finally, there were comments on the provision of task information. Breaking the student project into sub-tasks added a considerable degree of complexity and some students reported they found this hard to understand. For instance one student responded:

"Lack of information on time."

It's not exactly clear what they mean, however, it was a module design decision to release task information as and when it was needed and it may be the student was meaning he or she would have liked more of a big picture at the start.

(iii) Other dislikes (2 students)
Comments were made about the timing of lectures (these were scheduled 0900-1000).

5. SUMMARY
As we indicated at the outset, this paper is not a rigorous scientific study. We have not conducted randomised controlled trials. We not only introduced group projects but we also made significant changes to the curriculum. We cannot be certain that our observations can be solely attributed to the group work. Moreover, there is a lack of objectivity, since the authors were intimately involved in the design and delivery of the group project module. Nevertheless, we are sufficiently encouraged by the changes that seem to have been wrought, to add our experiences to those of other educators before us.

In summary, there are two findings. Group projects can be powerful agents for change. We have seen much improved levels of engagement by the students and, in our experience, unparalled levels of coursework submission and submissions by the deadline. Second, the devil really does lie in the detail. Communicating effectively with the large numbers of people is important. Designing an appropriate task (particularly in terms of getting the balance between the challenge and the manageableness of it) is important. And most important of all, careful allocation of students to teams.

ACKNOWLEDGEMENTS
The authors would like to thank all their colleagues and students for their enthusiastic support and tolerance of the inevitable teething problems.
REFERENCES


APPENDIX - Overall Description of the Assessment

Students will be expected to participate in each major class of activity, so for example, avoiding programming will not be permitted. This will be achieved by making Learning Outcome 4 mandatory. So each team member will be required to carry out a small, unseen maintenance task (T4) to their system in a controlled environment. It will be assessed on a pass/fail basis as it carries zero credits.

The entire team is expected to meet with their Tutor regularly (usually weekly but refer to the Module Calendar). Your Tutor is there to provide advice, guidance and encouragement but not resolve detailed technical questions which should be resolved during the Lab Sessions. Your Tutor will be the best person to ask questions about the running of the Group Project.

Tutors are also responsible for assessing Learning Outcome 1 (understand and apply the principles of professional and ethical behaviour in a group context). This means we expect you to be a reliable and committed member of your team. If you consistently fail in this regard you will receive a warning. However, you cannot pass this module without satisfying this Learning Outcome no matter how great your technical brilliance.

The Group Project comprises six individual tasks. Detailed information for each task is according to the calendar below. Note also that some tasks are to be performed and assessed as a group whilst others are individual.

At the end of the project each group will submit, if possible, an agreed distribution of relative contributions to the overall group work achieved. This distribution serves as a recommendation to the module leader. If a group is unable to agree the mark
distribution then each member will be asked to independently submit their own proposed distribution along with their log books to their Tutor for arbitration. Group marking will be on the basis of the individual mark = group mark + personal adjustment (if appropriate) where the personal adjustments must sum to zero.