How to prepare and support students for work with clients

1 Supporting/ facilitating team working

1.1 Team Selection

There are many different strategies for arranging a student cohort into project teams, a number of which we have experimented with over the past ten to fifteen years. In part, the strategy is determined by how much we know about the students and how much they know about each other. Creating teams in a cohort of new conversion Masters students
is, of necessity, an act of social engineering: we barely know the students and they have never met each other before. In such a context it is only feasible to be firm and allocate teams at random.it is necessary to help break the ice and some sort of social event can do this. We have found that most students will quickly get on with the job,
their life is so full of new activities and the pace of the courses so swift that they have no time to think about potential problems with the members of their group. Within a few days they have met their client and they have to produce something for the meeting with the manager that week. Student who have problems at this stage tend to have problems throughout and may well be unsuited to such a demanding course.

It is rather different with our second year students, who will already have worked in a tutorial group on a weekly basis from the start of their degree courses and who will have worked together on an internal team project, the Crossover project, in the first year. We could impose the composition of each team, with a view to distributing the most able students amongst teams, or the female students or whatever selection criterion we felt was most significant. However, we don't do this. We let the students choose their teams themselves.

This selection strategy surprises some people. They do not feel it fits real world practices. In industry, a manager would decide who to put in a team, based on some mixture of demonstrable skill and assessment of personalities.

We justify the students having a choice of team members in much the same way that we justify them having a choice of project client. We need some, if not all, of our teams to succeed, for the students' sake, for our sake and for our clients. Hence, we need to minimize the risk of project failure through failure of teamwork. Allowing friends to work together does seem to reduce the incidence of teams and team communication breaking down completely.
Furthermore, the element of team choice may give students a greater degree of ownership and sense of stake in the project, which is also likely to increase team commitment.

It is not hard to anticipate that there are problems with giving students choice. When friends in a team fall out, over differences of opinions about designs, about coding techniques, about team roles and effort required, this can be a lot more acrimonious and personally damaging than when team members who do not know each other come to blows. We don't see this happen very often, but when it does, project supervisors may have to display considerable tact and diplomacy in handling the fall-out.

Perhaps a greater difficulty is knowing
how to handle the students who do not want a choice about joining a team, who want to be told whom to work with. Usually, about 10% of the second year student cohort fall into this category. If we were to generalize about these students, we could identify some of them as the weaker students (rather like those of us who were never picked for the football or the rounders team, because people knew we could not control the ball); some of them are bright students whose preference is to work on their own and some of them are unmotivated students who have not bothered to make any effort to seek out a potential team. Most commonly, one or two teams will comprise students who have been placed together by the project organizer. We have not analyzed the performance of these teams quantitatively, but the strong feeling is that they struggle to get reasonable marks. It is possible that they do worse than if we had socially engineered team membership for the whole student group.

1.2 Team roles

Even when we give students the choice of whom to work with, we still advise them that they should choose team members judiciously and that they should look for a mixture of skills and strengths. We send an E-mail to all the students a few days before the course starts with the following suggestions: "Bear in mind, when organising
yourselves into teams, that most teams require a blend of skills, good programming knowledge, good organisation
and planning skills, good documentation skills, good skills at talking to the client etc. try to balance your team so that you can cover all these attributes. You will have to meet outside the lecture times so that is another thing to bear in mind."

Both on our second year "Software Hut" and conversion Masters "Maxi" project, we expect a team to appoint a leader and a team secretary. We further expect that these two positions will rotate around the team, so that each member has the experience of managing the group for a stage of the project and each has the experience of note taking and group document management. Our industrial project manager for the "Maxi" project, Stan Price, places a great deal of responsibility on the student as project stage manager. He or she is expected to collate all the team statistics and present figures for effort and progress at the end of the stage.

Of course, asking each student to take a turn at team leadership and management is helpful in assessing individual student performance (see section 7.2). It can presuppose that a project sticks to a waterfall lifecycle pattern, which in reality is not likely to be the case. Furthermore, in a short, semester-long project, the project stages are themselves so brief that the influence of the leader on team performance may not be discernible. Even in a two semester project like "Maxi", there may be a whole range of personality and inter-personal factors that determine the effectiveness of each individual as team manager, so assessment of performance of team roles must be conducted very carefully using evidence from a wide range of sources.

1.3 Team building and communications skills

So many times in the press we read criticisms of technology graduates and their paucity of team and communications skills. To a great extent, these skills can only be acquired through practice. So, what better justification for running some team activities on a computing curriculum! Many of the students, however, are uncertain about the prospect. Whilst they tell us that they value the real-life aspect of working for an industrial client, the majority also say that they would prefer to work alone.

Amongst our second years, we see many groups of students getting on with their industrial project as friends, supporting each other and sharing a joke, but we also see teams who do not know each other and who need some immediate support to start communicating together. After all, within a week or two of getting together, they will have to function as a team in front of an external client. So, to help ensure that teams who do not know each other well are not placed at a great disadvantage, we introduced specific tutorial sessions on team working at the start of the second year "Software Hut". We hope that these sessions are also helpful for the teams that are comprised of friends, that they too will learn from reflecting on the strengths they bring to a team and the areas in which they are reliant on other team members.

You do not necessarily have to run sessions on teambuilding and communications skills yourself. Few computing lecturers will have the experience of teaching such generic and personal skills. We enlisted help from our university Careers Service, who run and repeat an introductory two hour session for our second year project students, working with them in sets of approximately thirty students. If your Careers Service cannot help then your Staff Development Unit might be able to put you in touch with a suitable tutor.

We are at the stage now where we need to assess whether these teambuilding sessions are genuinely helpful. We also feel the need to do more to support students with extra practice in interviewing skills.

Undergraduate students with limited work experience may have little experience of planning and conducting a structured interview. It is worth considering running some role playing sessions in which students experience the roles of both interviewer and interviewee and have the opportunity to reflect on comfortable interview techniques and successful information gathering.

Our fourth year students in the student software company, "Genesys Solutions", have specifically asked for help and practice in negotiation skills, for which an experienced careers trainer was brought in. These students are expected to negotiate a fee for their work and find that difficult to do. However, even without a fee, the process of specifying, designing and building any software system is one in which compromises have to be negotiated between client and developers, so this is a useful skill that could be practised on team project courses at any level of study.

2 Setting expectations of students' work and attitudes

It is always easier to do something a second time round. After you have been running industrial projects for a few years, they begin to gain a reputation amongst the students: anecdotes about spectacular project successes and failures become part of the department's folklore. Indeed, your reputation for industrial projects might become such that students use this feature as a criterion in selecting your institution's degree courses. Once a course has a reputation, students will already have a firm expectation of it before they start. With industrial project work, it certainly helps if that reputation is a tough one! There's no excuse for being late! You sweat blood on this project! There's no excuse for not doing the work!

2.1 Expectations of professionalism

On both the second year undergraduate "Software Hut" project and the conversion Masters "Maxi" project, although the first is managed by lecturers and the second by an external industrial project manager, both tend to take a tough line in the first few weeks, but ease up later on, particularly with students who take our demands very seriously and who may find the pressure of project work very stressful.

In the main, we do not need to set high expectations to motivate the students. The clients' expectations are usually sufficient motivation. We do, however, have to reinforce expectations about professional behaviour, about time keeping, courtesy and confidentiality. Put simply, whilst non-attendance and late attendance at other lectures may be tolerated, lateness or absence from team project activities will be noted and marked down. It is made clear to students from day one of the project that each one is responsible for his or her own
efforts and for the success of the team. Things will go wrong with the project and part of the learning curve is coming to terms with problems as they arise and working out alternative solutions, rather than apportioning blame and giving up. The students are already aware of the concept of the "personal software process" (ref) and it is useful to remind them that the project is an opportunity to put this process into practice and find out what it means for real.

2.2 Expectations of software quality

It also helps students to set standards for the production of software documentation if they can look at and discuss concrete examples. This can prove difficult, because most textbooks only include small fragments of system documentation, contributing to a shortage of realistic case-study software development materials. This is one area in which a good working relationship between a department and a large software company might be an asset. You may be able to persuade the company to supply example documents, suitably anonymized. That is, if you are happy that the company's methods and quality standards complement those used throughout your degree and discussed in the lecture theatre.

One advantage of having run industrial projects for a number of years is that we can select some of the best examples of previous requirement specifications, test specifications, user manuals and other documentation and pass them round as illustrations of both the quality and quantity we are looking for....

say something more about changes in methods, difficulties students have faced in choosing appropriate modelling techniques...and the way our views have changed on the amount and nature of documents.

3 Setting time-scales for students' deliverables

The biggest risk to industrially focused student projects is that they will not deliver a successful outcome within the time-scales of the academic semester or year. In sharp contrast with real world commercial projects that frequently run late, albeit with the payment of penalties, educational projects must deliver work that can be assessed in time for examination boards, as well as work that constitutes a reasonably useful product for the client. There can be no exceeding the end date for the project.

This pressure to deliver is very obvious to us as project supervisors, but may not be immediately apparent to the project students, at second year or conversion Masters level. Left without guidance on deadlines for deliverables and project stage completion, most teams would get off to too slow a start and find it very difficult to make up lost ground in the last weeks of the project. Whilst we do want our students to learn about time management, resource planning in a practical environment, except at the level of the fourth year student company, it is too risky to leave all the decisions to them. We provide a week by week framework that suggests where we expect them to be in terms of project activity and percentage of total project effort completed. This framework is also useful, because we use it with our external clients when negotiating their involvement. This establishes a common agenda and a set of common expectations that students and clients may have of one another.