VICTORIA UNIVERSITY OF WELLINGTON
Te Whare Wananga o te Upoko o te Ika a Maui

School of Mathematical and Computing Sciences
Computer Science

PO Box 600
Wellington
New Zealand

Tel: +64 4 463 5341, Fax: +64 4 463 5045
Email: Tech.Reports@mcs.vuw.ac.nz
http://www.mcs.vuw.ac.nz/research

Supporting and evaluating team dynamics in group projects

Judy Brown and Gillian Dobbie

Technical Report CS-TR-99/7
April 1999

Abstract

Computer science students benefit from working in teams. But working in teams is difficult and team skills are seldom taught. In this paper, we describe mechanisms we put in place for supporting team processes in our group project course. We evaluate the mechanisms and extract guidelines that are useful for supporting team dynamics.

Publishing Information

This paper was published in the proceedings of the 30th SIGCSE Technical Syposium on Computer Science Eduation, 1999, pages 281-285.

Author Information

Both Judy and Gill have been involved in teaching Software Engineering courses at various levels in our department over the past few years. Both authors have other publications in this area. You can contact Judy and Gill via email: {judyb,gill}@mcs.vuw.ac.nz.

1 Introduction

Team projects are often assigned to computer science students in their senior years. These frequently occur within software engineering courses [1, 10, 14] but are also assigned in other courses such as HCI (human-computer interaction) [3, 12], database [14] and multimedia [5] courses.

Team projects have many benefits. They are seen as motivating to students. With a team project non-toy applications can be assigned by the lecturer. Team projects are also a great learning experience because they provide students with valuable experiences (developing abilities to work in groups, to respect the work and skills of others, to develop presentation and interaction abilities etc.) [7]. Most importantly though, team projects are a good preparation for industry projects. In industry, most software projects are not individual efforts but are accomplished by teams of qualified professionals. This is partly because of the size of software projects but also because of the link between teams and performance: "teams outperform individuals because they bring together complementary skills, create a situation where problems are solved more quickly, provide a social framework for working, and create a fun atmosphere in which to work" (p18, [9]). Software engineering books on project management also support this view [8, 11].

Computer science students have difficulties working in teams because they often do not experience team projects or group work in their first few years of study. Team work in foundational courses is unusual [6, 13]. In addition, most courses incorporating team work do not pay particular attention to teaching team processes or supporting effective team interactions. Nor do these courses evaluate or monitor the quality of the team dynamics.

This paper describes some guidelines for supporting effective team interaction skills for student projects that are in line with software engineering texts written for teams in industrial settings [4, 8, 11]. We also describe two methods for evaluating team interactions in group projects. We were motivated in this work by the difficulties our own students were having working in teams. We observed that a successful end product did not necessarily mean our students had had a positive team experience.

In this paper we describe our software engineering course in detail. In section 2 we describe the course, emphasising the aspects of the course that we feel were significant in supporting students to work effectively in teams. This includes both course infrastructure (we call these indirect mechanisms for supporting team interactions) and methods we used to teach team processes to students (direct mechanisms). In section 3 we discuss our method for evaluating team processes. In sections 4, 5, and 6 we discuss the results of applying these methods in our course. We conclude in section 7, with guidelines we extracted from our evaluation.

2 Team support mechanisms

In this section, we list the mechanisms we put in place to support the team process.

2.1 Indirectly supporting team processes

We indirectly supported the team process in the following ways:

- providing technical support: both in the form of school programmers and a tool assistant. The programmers ensured that the programming environment was operational and both the programmers and tool assistant gave tutorials. The tutorials were carefully timed so that tools would be introduced when students required them.

- defining process guidelines: the process guidelines were stated clearly at the beginning of the course. The guidelines included dates on which major documents were due and a marking scheme. The marking scheme rewarded individual contribution to the team process encouraging all team members to contribute to their team.
- defining and establishing a team structure: each team was assigned a client and a supervisor. The client is an expert in the project domain and their role is to clarify the project requirements and to resolve ambiguities as they arise. The supervisor is a mentor and their role is to guide, motivate and provide feedback to the team. Teams were required to choose a team leader.
- forming teams: a questionnaire that asked students to list their preferred team mates, skills, and work habits was used in team formation.
- assigning projects to teams: students select from a list of projects. Having students select from a list of available projects, rather than allowing students to propose their own project, helps students get down to work more quickly.

  The projects usually require students to develop software which is useful to someone in our school. Teams are more motivated if they are working on a project they are interested in and if the resulting software product has potential real use.

  We allowed more than one team to work on the same project so no team was given preference over another when projects were allocated.

2.2 Directly supporting team processes

We directly supported the team process in the following ways:

- learning from past students: Teams can also learn from the experience of others, especially when that experience is very close to their own. Annually, students are asked to comment on "Managerial Lessons Learned" in an essay they write at the end of the course. We put the essays on the web and ask current students to read the comments of past students.
- providing tutorials on team processes: Teaching students about the team process makes them aware of some of the difficulties in team work.

  We provided a tutorial on teams which was divided in two parts. The first part summarized the positive points made by previous students in the "Managerial Lessons Learned" document. It covered how to set realistic project goals, wisely allocate tasks to team members, run meetings, manage time, and communicate and manage shared group documents (like meeting minutes and design specifications). We also described the roles of the team leaders and the team members.

  The role of the team leader was to :

  - coordinate the activities of the team (tracking progress, scheduling work)
  - motivate the team
  - ensure the team communicates effectively
  - interface with supervisor, arrange meetings with client when necessary
  - set agendas for meetings.

  Possible additional roles for the team leader were: meeting facilitator, minute taker, helper, and allocator of tasks.

  The role of the team member was to:

- help to set the team goals (project goals, task allocations)
- help the team move towards these goals
- accomplish tasks given to them
- meet deadlines
- attend team meetings
- contribute to developing a productive atmosphere within the team.

A clear definition of roles clarifies responsibilities and gives a minimum of authority to team leaders.

The second part of the team tutorial was given by an invited speaker who had extensive experience in teams in industry and academia. His talk was entitled "When Groupwork doesn't work... What to do about it". As well as addressing problem areas, the speaker talked about methods for creating 'energized' groups. This lifted the discussion about the team process away from problems to rewards.

3 Team Dynamics Evaluating Methods

In order to assess how well the mechanisms supporting the team process worked, we used the following two techniques:

- surveys: we administered one survey at the mid-point of the course and one at the end of the course in order to assess how well we had supported the team process at the beginning and throughout the course.
- an essay: students comment on "Managerial Lessons Learned" in an essay they write at the end of the course.

In this paper, we briefly summarize the results of the first and second surveys and discuss the essays.

Many of the questions in both surveys asked students to rate items (such as a talk) on a standard Likert scale of 1 to 5 where 1 was 'not useful at all' and 5 was 'very useful'.

4 Mid-course survey results

In [2] we documented a preliminary evaluation of our team support mechanisms based on the first survey in week 7 of a 12 week course. Twenty-two of the twenty-six students in the course completed the survey.

This survey showed that students felt we had supported team processes well at the beginning of the course. It confirmed that our students were very inexperienced with team work in the university context. In addition, all students reported being basically happy in the team they had been allocated, which confirmed for us that our team formation methods were working well.

4.1 Assessment of direct support

Students appeared to enjoy learning from the experience of past students. The essays "Managerial Lessons Learned" which we made available to students were widely read and rated as moderately useful.

The tutorial on the team process was rated as useful. Students particularly liked the role descriptions of the team leader and team members in this tutorial. These were rated as useful. We hypothesize that the clarification of responsibilities helped the teams to function by specifying fundamental responsibilities for these roles. The students found the second part of the tutorial on team processes, which discussed common team problems and creating energized teams moderately useful.

Technical tutorials rated similarly to process tutorials ranging from moderately useful to very useful indicating to us that students valued process knowledge similarly to technical knowledge.

4.2 Assessment of team skills

We also asked our students to assess their team skills. Individuals asserted that their teams had many skills. We listed 14 team skills. In all, one hundred and eighty-eight items were ticked by the students. This was an average of 8.5 ticks per student. More than half of the students in the class indicated their teams had the following skills: maintaining team spirit, allocating tasks to members, coping with conflict within the team, identifying essential functional requirements, communicating, being creative, running meetings, coping with difficult personalities (e.g. individuals who dominate), prioritizing and planning tasks, coping with team members with limited abilities, managing the project size, coping with slack team members, and getting tasks done on time.

Similarly we asked students about difficulties they experienced in the team process. We listed the 14 team skills from the previous question. Twenty-four items were ticked overall, an average of 1 tick per student. No items were ticked by more than half of the students. The item receiving the most ticks (8) was getting tasks done on time.

We interpreted the skills results cautiously and concluded only that there appeared to be a considerable amount of team pride amongst our students at the mid-point of the course and that the team process seemed to be working well so far.

5 End-of-course survey results

Twenty-one of twenty-six students completed the second survey which was administered in week 12 of the 12 week course.

In this survey we evaluated the students' impression of how well we had supported the team process throughout the course. We also asked students to tell us about ways in which we could further support team processes.

5.1 Assessment of indirect support

A section of the survey assessed various course organization methods we employed.

Our students found the major deadlines we set for the production of documents useful.

Our assessment scheme rewarded both team results and individual effort. The project component of the assessment is 70% of the final grade. This is divided into a group (30%) and an individual (40%) component. We thought that the group component in the assessment scheme would encourage students to contribute to their team. As it turned out, the assessment scheme was only moderately influential in shaping student behavior within the teams.

We provided support staff for the teams, namely the supervisor and client. Supervisors were rated as moderately useful in providing technical support. Clients were moderately useful in defining project requirements. The school's programmers were rated as moderately useful and the tool assistant was considered useful.

Each team had a Unix group, email address and web page allocated to them which they were encouraged to use to facilitate communication and sharing of information between the team members. Their Unix group was useful, the mailing list was useful, but rcs (revision control system) was not particularly useful. The skeletal web page we developed for each group was rated very useful. We introduced the web page as a tool to help students organize themselves and were surprised how useful the teams found this mechanism.

5.2 Assessment of direct support

We were interested to find what advice students took on board with respect to organizing their meetings.

Students uniformly reported meeting formally once a week. The formal meetings usually lasted one hour. We had encouraged students to take minutes and assign a facilitator. Eighteen of twenty-one students reported that they took minutes in their meetings. Seventeen of twenty-one students reported that they had assigned a facilitator for a meeting. We asked students what functions the meeting served for the team. The students reported that formal meetings were useful in organizing their project, developing strategy, "keeping their supervisor happy", discussing issues, "breaking down documents/programs", presenting problems, presenting ideas, getting information and feedback from the supervisor, and reporting to the supervisor.

The organization of the formal meetings varied greatly. Some meetings followed an agenda while others were free form discussions. One individual reported that his team used a consistent structure for their meetings. The following items were always on the agenda : the week in review, the week ahead, a check on whether everyone was happy, delegation of tasks, discussion of problems and then finally feedback from the supervisor. This was the agenda of the most organized team and not the norm.

A preferred strategy for accomplishing work was specialization. We had recommended this as a useful approach in our team process tutorial. Nineteen of twenty-one students said that group members had specialized tasks. Specialization did not necessarily mean that students worked individually though. Students reported that approximately 52% of the work they did was done independently and about 48% of the work was done collaboratively.

5.3 Assessment of team skills

We reassessed team skills at the end of the course to compare with the mid-point assessment.

One hundred and seventy-two items were ticked at the end of the year. This was an average of 8.2 ticks per student. The number of ticks per student had dropped only slightly from an average of 8.5 ticks in the mid-year, indicating that team spirits were comparable at the mid-point and at the end of the course.

The top four skills (ranked in order with the number of ticks indicated) follows:

- **20:** allocating tasks to members
- **16:** maintaining team spirit
- **16:** being creative
- **15:** identifying essential functional requirements

The number of reported difficulties in team skills had increased substantially from an average of one tick per student to 1.7 ticks per student (i.e. a total of 37 ticks). The project became more stressful as final deadlines approached. An increase in the demands of the project was reflected in an increase in the number of problems that arose within the team.

The following is the prioritized list of the top four most difficult areas:

- **7** getting tasks done on time
- **6** coping with slack team members

- **5** communicating
- **4** celebrating successes

Difficulties with coping with slack team members had not been ticked in the first survey. It was almost at the top of the list in the second survey. Either team members were not slack early on in the project or slack team members were more easily tolerated in the early part of the project.

More difficulties relating to communicating arose in the second part of the course. This item, which barely appeared in the first survey, now ranked number 3 in the second survey.

6 Summary of student essays

Students found our course to be rewarding but exhausting. These comments, from the essays "Managerial Lessons Learned" summarize the course experience for some students (both positive and negative).

> All the hard work pays off ... in the long term. ... I enjoyed the course very much and believe that a lot of that had to do with what the team and I made of it and put into it.

> [This course] will probably be the most stressful, headache inducing, time consuming, sleep depriving, rewarding, interesting and useful course you will ever do.

The course was valuable to the students in many ways but students often cited the team experience as being the most rewarding part of the course.

> I have learned a lot from doing this course. Most of all, I learned a great deal about the management aspect of team work.

> The primary value of this course is not in the credits at the end of a semester. Its value is reflected in the exposure you receive to software project development in a team environment.

Almost all students made some comment about the value of having a team leader. This occurred even in a team where the team leader did not show strong leadership abilities. None of the students reported that their team was dysfunctional.

7 Conclusions

We suggest the use of an essay as a useful method for evaluating team dynamics on an annual basis. Detailed surveys can be employed to further evolve or change a course. Both methods provide valuable information on how students experience team work.

From the results of the evaluations described in this paper, we have derived the following guidelines for supporting effective team interactions. These guidelines are based on identifying team support mechanisms that were evaluated as useful or very useful by students.

Successful indirect support mechanisms include using a method to form the teams which takes into account student preferences, skills, and work habits. Providing challenging and novel projects from a domain in which the students have an interest helps to motivate the students. Setting regular deadlines and clearly specifying project outcomes helps students organize their time. There is also evidence that students prefer to work in teams with a leader. The leader provides a point of contact for the team and coordinates the work of the team. Having support staff (supervisors, clients, programmers, and the tool assistant) available to the teams genearally assisted them in their work. We feel the rating in the survey for each role was more dependent on other factors including the personality and ability of the individuals filling the role rather than the usefulness of the role itself. The provision of tools such as Unix groups and mailing lists helped teams organize their work. In particular, the web page which provided a simple framework for storing and accessing documents (meeting minutes etc.) was valued by the students.

The direct support mechanisms also proved successful. The information conveyed in the team process tutorials was appreciated. This included advice on how to run meetings, advice on specialization as a technique for accomplishing work, as well as advice on how to work in a team. Students also found a description of the roles of team leaders and team members useful. The tutorial mechanism appears to be a good method for transmitting this advice. Students found formal meetings to be valuable and also generally agreed that specialization was a useful technique to accomplish team goals. The essays written by past students was a useful resource to current students. The tutorials and essays helped students anticipate difficulties and reduced the number of dysfunctional teams.

We believe that learning to work in teams is a necessary skill for computer science students and we have shown that it is worthwhile teaching and supporting team process skills explicitly.

References

[1] Stefan Biffl and Thomas Grechenig. Teaching software engineering and project management to 300 participants without drain of quality or intensity. In *ITiCSE (Integrating Technology into Computer Science Education) conference proceedings*, 1997.

[2] Judy Brown and Gill Dobbie. Software engineers aren't born in teams : Supporting team processes in software engineering project courses. In *Software Engineering: Education and Practice*, 1998.

[3] Matthew C. Clarke. Teaching the empirical approach to designing human-computer interaction via an experiential. In *SIGCSE conference proceedings*, 1998.

[4] Larry Constantine. *Constantine on Peopleware.* Prentice Hall, 1995.

[5] Ainslie E. Ellis. Design, implementation and testing techniques for multimedia industrial experience projects. In *ITiCSE conference proceedings*, 1996.

[6] David Finkel and Craig E. Wills. Computer supported peer learning in an introductory computer science course. In *ITiCSE conference proceedings*, 1996.

[7] Judith L. Gersting and Frank H. Young. Content + experience = curriculum. In *SIGCSE conference proceedings*, 1997.

[8] Watts S. Humphrey. *Managing Technical People.* Addison-Wesley, 1997.

[9] Jon R. Katzenback and Douglas K. Smith. *The Wisdom of Teams: Creating the High-Performance Organization.* Harvard Business School Press, 1993.

[10] Michael J. Lutz and J. Fernando Naveda. The road less travelled: A baccalaureate degree in software engineering. In *SIGCSE conference proceedings*, 1997.

[11] Steve McConnell. *Software Project Survival Guide*. Microsoft Press, 1998.

[12] Margaret M. McManus. Gathering together through groupware in an MA program. In *ITiCSE conference proceedings*, 1996.

[13] Viera K. Proulx, Rasala Richard, and Harriet Fell. Foundations of computer science: What are they and how do we teach them? In *ITiCSE conference proceedings*, 1996.

[14] E.E. Villareal and Dennis Butlar. Giving computer science students a real-world experience. In *SIGCSE conference proceedings*, 1998.