Evaluating Groupware Support for Software Engineering Students

Sarah Drummond
Department of Computer Science
University of Durham, UK
Sarah.Drummond@durham.ac.uk

Cornelia Boldyreff
Department of Computer Science
University of Durham, UK
Cornelia.Boldyreff@durham.ac.uk

Magnus Ramage
Department of Computer Science
University of Durham, UK
Magnus.Ramage@durham.ac.uk

Abstract

Software engineering tasks, during both development and maintenance, typically involve teamwork using computers. Team members rarely work on isolated computers. An underlying assumption of our research is that software engineering teams will work more effectively if adequately supported by network-based groupware technology. Experience of working with groupware and evaluating groupware systems will also give software engineering students a direct appreciation of the requirements of engineering such systems.

This research is investigating the provision of such network-based support for software engineering students and the impact these tools have on their groupwork. We will first describe our experiences gained through the introduction of an asynchronous virtual environment – SEGWorld to support groupwork during the Software Engineering Group (SEG) project undertaken by all second year undergraduates within the Department of Computer Science. Secondly we will describe our Computer Supported Cooperative Work (CSCW) module which has been introduced into the students' final year of study as a direct result of our experience with SEG, and in particular its role within Software Engineering. Within this CSCW module the students have had the opportunity to evaluate various groupware tools. This has enabled them to take a retrospective view of their experience of SEGWorld and its underlying system, BSCW, one year on. We report our findings for SEG in the form of a discussion of the hypotheses we formulated on how the SEGs would use SEGWorld, and present an initial qualitative assessment of student feedback from the CSCW module.

1. Introduction

In recent years there has been an appreciation of the benefits that can be obtained by software engineering students working in groups or teams (Bannon & Schmidt, 1991; Boldyreff, Drummond, & Walker., 1997; Brereton et al., 1998; Gotterbarn & Riser, 1994; Habra & Dubois, 1994; Harrison, 1997; Robillard, 1998). As well as reinforcing theoretical concepts, group projects provide students with experience of the type of teamwork found in industry. It is important that students gain experience of this mode of working and that they are provided with appropriate computer support.

CSCW has been well reported in European and international conferences for the past fourteen years, and has formed an important background to our research reported here. However, little research has been reported on the development of CSCW to support groupwork in higher education and, in particular, software engineering education. Nor has the topic of CSCW applied to software engineering been extensively addressed by current research; exceptions

are the work reported by Grinter (Grinter, 1997; Grinter, 1999). In contrast there has been more research undertaken in the evaluation of groupware but this has been from a more business oriented rather than an educational perspective (Grudin & Palen, 1995; Mark, Fuchs, & Sohlenkamp, 1997; Orlikowski, 1992). The remainder of this paper addresses the educational context of our research in section 2. The Software Engineering Group (SEG) project is described in section 3, followed in section 4 by the network-based computer support developed which is based on the groupware tool "Basic Support for Cooperative Work" (BSCW) (GMD-FIT). The results obtained to date for SEG work are presented in section 5. Section 6 introduces the structure of the CSCW module and the criterion used for student evaluation of four groupware tools. Section 7 addresses our initial findings from the CSCW module, which are based on student feedback from a questionnaire with emphasis placed on BSCW responses, enabling us to make a comparison between the usage in the SEG year and subsequent CSCW year. A final section presents our conclusions.

2. Software Engineering education at Durham

The Software Engineering (SE) module is taught to all 2nd year undergraduate students studying in the Computer Science department. An important part of the module is the practical component that consists of the Software Engineering Group (SEG) project that runs in parallel with the SE lectures throughout the academic year. The lectures in SE cover all the major concepts relevant to the software lifecycle activities as well as topics relevant to the management of software projects.

In the SEG project, students carry out all of the main activities of the software lifecycle supplemented by intermediate tasks undertaken as supervised practical work. The intermediate tasks include e.g. introduction to desktop video conferencing, introduction to the shared workspace, domain analysis, cost estimation, project planning, risk management, groupwork, configuration management.

The students carry out the majority of the SEG work independently in small teams. Each team has a member of staff who acts as the group's tutor, consultant, and customer. Typically students will meet with their tutor fortnightly to discuss their progress.

Most students taking a degree within the department go onto study SE II in their final year. These students undertake an individual final year project (which always involves a major implementation) but the final year modules rarely involve any team work. Thus, the SEG project described in more detail below, is the students' main experience of teamwork based development during their degree course

In addition to students choosing the SEII module many take the CSCW/Requirements Engineering module. The CSCW component exposes students to practical usage of groupware and to the related social, organisational and design issues relevant to CSCW as well as the role of CSCW within Software Engineering. Within this module students are asked to evaluate four groupware systems. Experience that these students have gained from undertaking SEG the previous year allows them to take a retrospective view of BSCW in particular. The web pages developed as a resource to support students taking this module give fuller details of its content (Boldyreff,).

3. The Software Engineering Group (SEG) project

SEG projects have run successfully since 1984 within the Department of Computer Science. Their introduction and subsequent development has been largely motivated by a perceived need to prepare students for typical working practice found in industry. This type of project presents the first opportunity for the student to work as part of a group, to divide up work among several team members and make technical and managerial decisions as a group - a not uncommon real-life parallel.

The project itself is well structured into phases (Drummond, Boldyreff, & Munro, 1997) (Figure 1), and follows the classical waterfall software lifecycle model, with some optional prototyping. The waterfall model, generally implies that software development is undertaken in a series of definite steps, with no iteration, whereas in reality, software development can be carried out in parallel and iteration is common. Within the SEG project work, iteration is provided for, by allowing the students to specify changes at the beginning of each phase. McDermid discusses this type of iterative interaction in (McDermid & Rook, 1991).

Students have the opportunity to evaluate the work of other groups as well as their own; for example, at the end of the requirements phase each group carries out an appraisal of another group's requirements document followed by acceptance testing for the product developed by the same group. At the end of the group work, the all students are asked to produce a project legacy report where they take a critical look at how they have worked as a team during the project, what went wrong, how they rectified it and finally would they do it differently next time. This is a valuable learning exercise for the students. These legacy reports provide valuable feedback to all the staff involved in the SEG project.

One of the major achievements of the SEG project is that it provides students with early experiences with system building concepts and practices. This meets the industrial need for graduates with experience of building systems in a team rather than experience of simply working as a collection of individual programmers as discussed by Goldberg (Goldberg, 1998). In recognition of this, there are various industrially sponsored SEG prizes awarded each year.

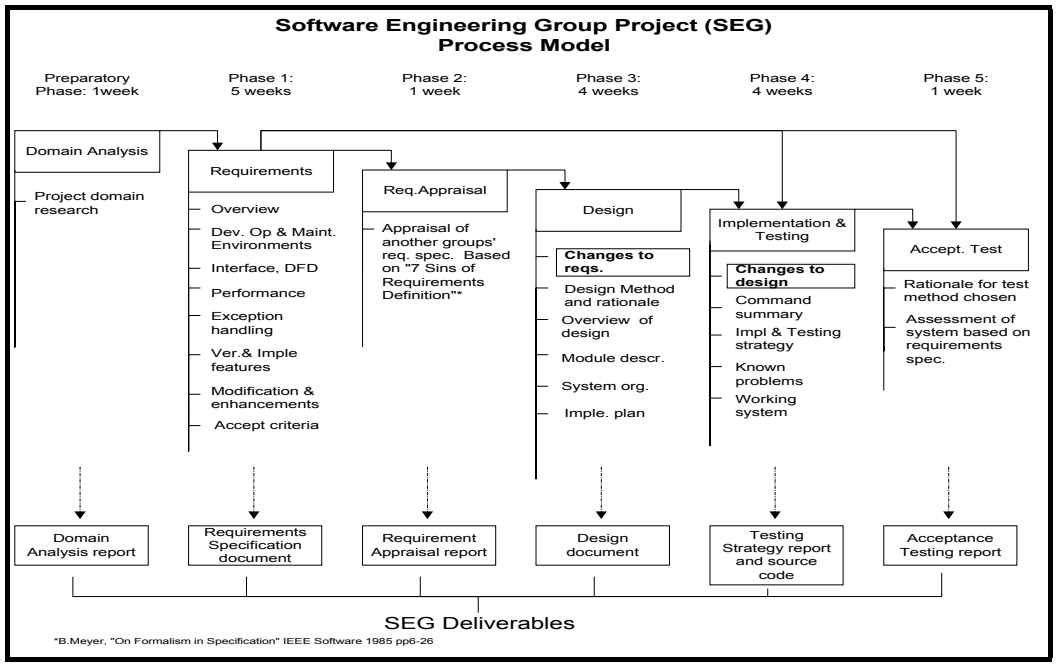

Figure 1: SEG project phases and process model

Having successfully established and run SEG projects, the department has considered how these could be improved to more realistically mirror industrial practice in software engineering. In particular, these considerations have given rise to studies to identify appropriate groupware technology for supporting SEG projects and the development of a virtual software engineering environment. With university funding we have been able to develop and monitor the introduction of network-based asynchronous computer support for the SEG project (Boldyreff et al., 1997). We now are able to report on our experiences of developing and using the virtual software engineering environment since 1997.

4. Creation of *SEGWorld* - a virtual environment for Software Engineering students

Our initial studies (Layzell et al., 1998) identified a need for groupware tools to enable SEG students to easily share documents and applications; we therefore investigated how we could effectively introduce an asynchronous groupware system - BSCW into the SEG students' working environment.

In the initial phase of the development, a virtual environment, *SEGWorld*, based on BSCW has been developed. *SEGWorld* is Web-based and essentially provides a repository for all the relevant teaching materials associated with SEG projects. A public workspace is provided, which allows all SEG students and associated staff, access to software tools relevant to student project work, and to other facilities i.e. posting general notices or queries. Private group workspaces, allow for the development and secure keeping of each group's practical reports and project deliverables.

To further support students during the SEG project, the Department of Computer Science funded a small multimedia PC laboratory dedicated for SEG student use. To better utilize the SEGLab, we have divided this space into three small offices and a common meeting area. A

Web-based online booking form has been developed which permits groups to book an "office" and allows staff to monitor SEGLab usage.

During the development of *SEGWorld,* we formulated a number of hypotheses about providing groupware, i.e. how it would be used, how it would support the students, and its importance to their work. In the following section two of these hypotheses will be discussed together with an overview of the supporting evidence obtained during a full year of trial usage i.e. 1998-1999[1] (Drummond & Boldyreff, 2000)

5. Results obtained during the SEG trial year

Throughout the academic year, of 1998-1999 the students' usage of *SEGWorld* has been monitored and data collected. The data collection methods chosen for this research are, in the main, observational, questionnaires and project monitoring.

This project monitoring took the form of BSCW automatically generating and emailing a list of activities undertaken each day in the group workspaces. This information includes the type of activity, student name and time. In addition to this, students were asked to complete questionnaires, and invited to take part in focus group discussions in order that their views could be collected.

In total there were 72 students in groups of 6 or 7 (12 groups). Questionnaire results are based on responses from 58 students (the completion of questionnaires was not mandatory).

The following hypotheses have been selected for discussion:

1. The introduction of an asynchronous shared workspace into software engineering groupworking will aid group members to organize and coordinate their work.
2. Greater use of shared workspace functionality will be made as the project progresses

The following subsections present each of the hypotheses and the discussion of the associated results.

Hypothesis 1

The introduction of an asynchronous shared workspace into software engineering groupworking will aid group members to organize and coordinate their work.

From a high level perspective, figure 2 represents the responses from individual SEG students, related to the workspace enabling better organization and coordination of their work.

[1] A more detailed description of this work and results obtained can be found in Sarah Drummond's MSc thesis (Drummond, 1999).

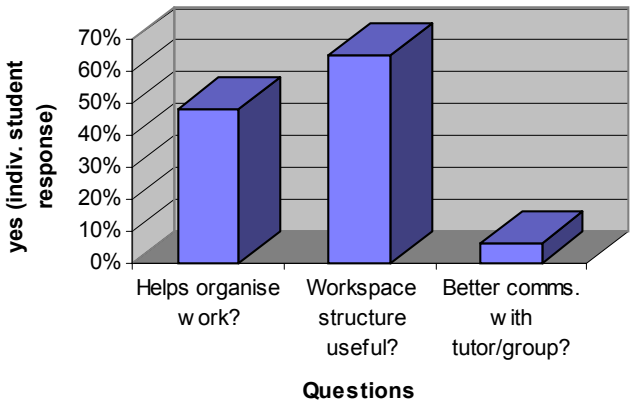

Figure 2: Organization & Coordination activities

In general, the students felt that the hierarchical structure of the workspace was intuitive and graphically illustrated how their work was being structured. But, as the level of decomposition of folders (directories) into sub-folders (sub-directories) increased, navigation became slow. Students commented on the lack of shortcuts to the various documents. In fact, students were simply unaware that shortcuts are possible. As SEGs have a group UNIX account in addition to their private workspace, five of the groups used both, with UNIX generally being the preferred choice because of faster system response times. The poor uptake of the communication functions within *SEGWorld* (i.e. email and automatic meeting facility), was due to the fact that groups met with their tutor face-to-face on a regular basis, both formally (arranged meetings) and informally (i.e. at the end of a lecture), to discuss progress and/or problems.

From a lower level of granularity, figure 3 highlights a selection of functions provided by the system, and indicates their usage as reported by the students.

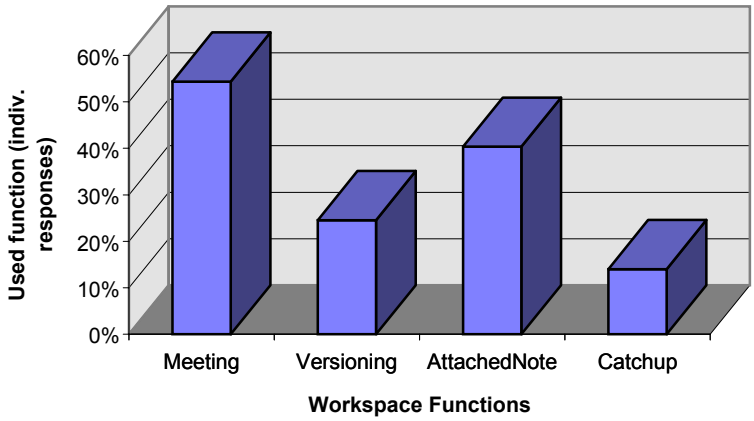

Figure 3: Functions relating to organization and coordination

These functions have been chosen because they are associated with organisation and coordination. They are as follows:

- *Meeting* - schedules a new meeting showing venue details and those invited to participate. E-mail is automatically generated to inform members of these details.
- *Versioning* - versions a document. A new version is created which becomes the current version, whilst old versions are still readily available.
- *Attached Note* - attaches a note to a specified object that is displayed to other users when they attempt to access the object. There is no formal locking mechanism for objects provided, when removed for editing etc.
- *Catchup* - A new document/object has a "NEW" icon attached. This "NEW" icon remains regardless of how old the document/object is, unless the catchup facility is used to remove it. This distinguishes new documents from existing documents.

Whilst the meeting facility was thought by group members to be useful, many did not use it because it was simpler to use existing e-mail systems. To organize a meeting via *SEGWorld* involved loading a browser, entering *SEGWorld* and then the group private workspace. To confirm attendance at the meeting involved every attendee replying in this fashion. Students were asked if they had used this function and whilst figure 3 indicates that over 60% of group members had, this figure does not reflect the actual low usage over the phases of the project. Many of the students had experimented with the meeting function during the initial *SEGWorld* tutorial session, but did not use it to any great extent after this.

The versioning mechanism provided was easy to apply, but few of the groups used it. An interesting point noted in the results obtained via the questionnaires were that within at least two groups, all members stated they had used this function. When these results were checked against the automated daily activity logs, it was found that only two members from each of these groups were shown to have actually used the function. This anomaly may be due to inaccurate completion of questionnaires, or that the group members worked around one PC. Within all groups, one member was appointed as secretary, and often this role involved controlling the versioning of documents.

The catchup function, which provides an up-to-date view of the activity i.e. new document, which has occurred within the workspace, was used very little. On further questioning, most students admitted to not being aware of what this function actually did.

SEGWorld provided a central repository for all group documentation, and as such provided a graphical representation for configuration management (i.e. a historical trail for each document), and awareness of other group members activities, i.e. determining if a group member had produced or read a section of a document. This in itself helped the groups in coordinating their work by being aware of the status of a document. From an organisational viewpoint, the workspace provided each group member with some insight into the contributions being made by other members, but much of the organisational strategy developed (e.g. distribution of tasks) was in the main, undertaken through face-to-face communication.

Hypothesis 2

Greater use of shared workspace functionality will be made as the project progresses.

The following graph (figure 4) shows the use made by SEGs, of the various functions provided by *SEGWorld,* during the different phases of the software lifecycle. These functions

are a subset of those available, and were chosen as they represented the most common events that would occur in the process of producing a typical SEG project deliverable.

The objective of logging the daily usage of these functions was to determine if the use of *SEGWorld* increased as the project progressed. This anticipated increase could indicate that the students were becoming more confident in using the workspace, and had overcome any initial problems.

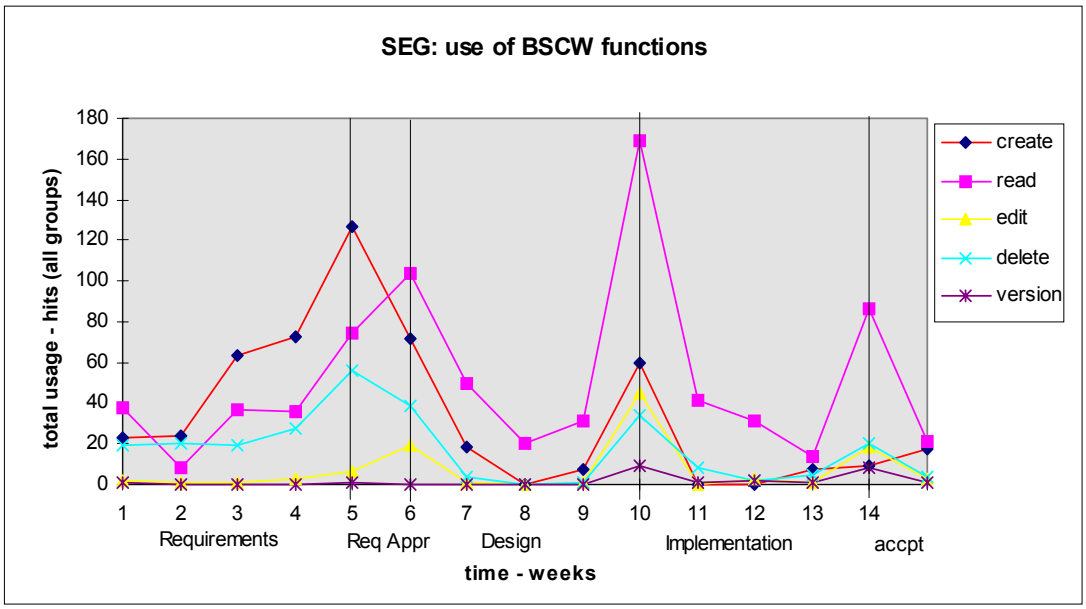

Figure 4: Usage of SEGWorld functions during the phases of the software lifecycle

In figure 4, most activity is centered on creating and reading the requirements specification. The negligible amount of activity by most SEGs for the editing and versioning functions could indicate that they did not fully understand these functions. Rather than editing or versioning an existing document it would appear that they have deleted and then re-created the document. At this early stage in the use of the workspace this was not unexpected.

The requirements appraisal phase (figure 4) shows the use of the create and read activities being high. In the case of the create function, approximately 50% of the usage was from three groups only. The edit function has begun to be used. This phase is for one week only and the deliverable is a relatively short document.

The design phase (figure 4) of SEG is a work intensive phase. Within this phase it can be seen that there is a marked increase in the use of the functions towards the end of the phase. The edit function usage has increased whilst there is a decrease in document creation; this may indicate better student understanding of these functions. Whilst versioning has been used by most groups its usage was still disappointingly low.

The implementation phase consists of developing the product software and a report which details the implementation and testing strategy and any known problems with the system. A departmental decision was made at the onset of the SEG project, that the use of the *SEGWorld* for developing code, would be inefficient. BSCW is a generic tool and as such offered SEG no support for software code development.

What is evident from the above graph is that the use of the version function increased slightly as the project phase progressed, and more appropriate use was being made of the create, delete and edit functions.

Overall, utilization of some of the more useful functions, e.g. versioning, was poor and few students made use of additional functions provided. This has been attributed to the following factors:

- students were aware of many functions but were insufficiently motivated to gain an understanding of how to use them,
- at times, usage of the workspace was hampered by poor response times of the network,
- there was a mismatch between the work in the implementation phase and the support provided by the workspace e.g the use of Modula-2 imposed too great an overhead on SEGs, as all modules would have to be continually downloaded.

Of these factors the main problem that needs to be overcome is the students understanding of the concept of the shared workspace. Initially the students were introduced to *SEGWorld* via an online tutorial. Whilst figure 5 shows that students thought the tutorial was useful, and that *SEGWorld* was intuitive (hence the lack of time invested in learning the system), they in fact under-used the system because they did not fully understand the functions available to them.

This under-use has been attributed to certain factors some known i.e. hardware performance, and some that were revealed via student comments. One problem that students encountered was the slow response times of the system. This was a major contributing factor to the slow uptake of the continued use of *SEGWorld*. In addition to this, the introduction of *SEGWorld* (via a tutorial) was not at the onset supported by experienced postgraduate laboratory demonstrators, therefore simple problems that arose at this stage were left unsolved and the students formed a poor image of the system.

This problem of introducing a new concept such as a shared workspace highlight that it is not just a question of giving students a new application with associated tutorial. Grudin (Grudin, 1994) points out that it is not just the technical issues which must be addressed for the successful adoption of groupware. The shared workspace is more than just an application, it also provides awareness of other members activities, and requires the group members to organize and coordinate their work differently.

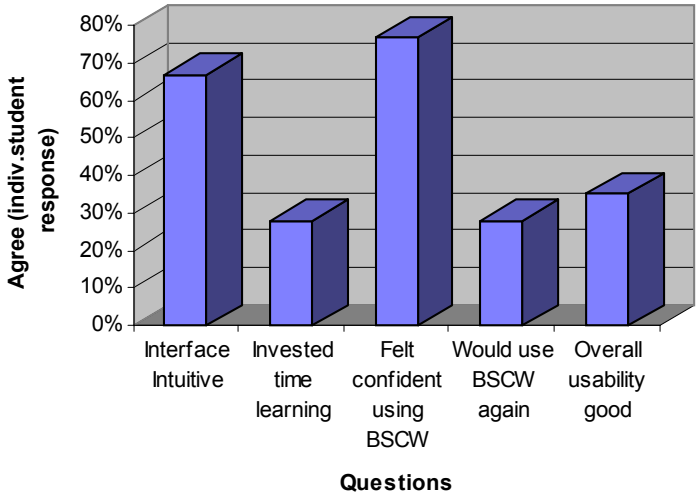

Figure 5: Students General Responses

The use of *SEGWorld*, despite the problems described above, provides the groups with a more stable and flexible environment than previously available. The students appreciate the benefits of the shared workspace but performance problems and a more focused guidance on usage are major issues that need to be addressed in the longer term.

6. Evaluation of SEG results

With respect to organization and coordination, the introduction of a shared workspace has provided a formal setting for the practical side of the software engineering module, and the students have found this both helpful and useful. Their use of the workspace functions related to these activities was reasonably comprehensive, but not consistent throughout the year.

SEGWorld has allowed us the opportunity to monitor student activity for the purpose of data collection for this study, thus providing the data with respect to functions used by the students. The students did not generally make greater use of the workspace functions as the project progressed, but did make more effective use of those they used. Their use of functions was generally steady up until the implementation phase. During this phase the support for code management was inadequate, and students simply used their accounts on the Novell network.

It is believed by the authors that the concept and use of *SEGWorld* was in principle, well received by both staff and students, but, in reality the system was under used. The reasons for this have been attributed to firstly, the poor system responses times which proved to be the largest complaint received from students; secondly, the initial introduction to the system was via an on-line tutorial which proved to be limited, and thirdly, the early *SEGWorld* sessions were not supported by experienced demonstrators. We have made some headway in addressing these problems in recent years as we have continued to use *SEGWorld*.

While using SEGWorld, the students do gain insight into the need for developing protocols for the way in which they worked i.e. how best to organize, structure and coordinate their work. A side effect of this is better group fusion. SEGWorld not only provides the students

with the opportunity to evaluate new technologies in a practical manner, but also allowes them to gain an understanding of group interaction, and how these interactions must be supported, which is a vital factor in the design of groupware. The students all feel that the experience gained of groupware is important to them for their future employment prospects.

Some unexpected benefits of introducing *SEGWorld* into the department's teaching have resulted. The visibility of the work of the SEG students reflected in the *SEGWorld* private workspaces, has provided the SEG tutors with some insights into the contributions of the individual students within the group, and more generally gives a means of assessing group's progress. This individual monitoring was not a primary requirement of the system but has proved a very useful feature. Another benefit is that both students and staff have gained practical experience of groupware requirements in a software engineering context. It has made both students and staff aware of the need for groupware to be designed and developed with specific support for software engineering tasks.

With the experience we gained within the CSCW field during our research on virtual environments in general and SEGWorld in particular, it was decided to develop and introduce into our curriculum a final year optional module on CSCW in 1999-2000. The SEG students monitored in 1998-1999 now proceeded to their final year and a number of them chose the CSCW option. We believe that it is an interesting exercise to follow these students from their use of BSCW during the SEG exercise to a more formal evaluation of other similar groupware tools including BSCW, as part of the CSCW module. The structure of the CSCW module and the evaluation criterion are described in the next section.

7. Introduction of a CSCW module

In the academic year 1999-2000, the success of SEGWorld as reported above was one of the reasons for the creation of a third-year (i.e. final year) module on Computer-Supported Co-operative Work (CSCW). This course was paired with one on Requirements Engineering (RE); the bulk of the RE part of the module was taught in the autumn term of 1999, and the CSCW part was taught in the spring term of 2000.

The CSCW module was designed with two distinctive features. First, it was intended to be interdisciplinary, and in particular to stretch the awareness of Software Engineering students on issues from the social sciences that they had not previously encountered. Thus two members of staff within the Computer Science department co-taught the module: a senior academic from the Distributed Systems Engineering group with considerable knowledge and experience in technical aspects of CSCW; and a junior colleague whose research area is in socio-technical issues within CSCW. Each lectured on issues from their areas of expertise. For such a broad subject as CSCW, which potentially covers so many areas, this seemed a useful way to achieve a sufficiently broad perspective.

The other distinctive feature of the module, of more interest to this paper, was the explicit use of groupware tools as part of the learning experience of the students. Students were divided into groups of 5 or 6 (with a total of 46 students on the module, this gave 8 groups) and were asked to spend a period of two weeks with each of the four tools to be evaluated. In each two-week period, the groups were set an essay-type question, relating to the current topics in the lectures, and asked to discuss this using the system. They were told that a portion of the

practical element of the module (which formed 15% of the marks for the whole module) would be based on the quality of individuals' participation in the discussions. They were therefore required to submit a transcript of the discussion at the end of each two-week period, and to find a way of recording the discussion so that this was possible (this varied with each tool). They were also required to construct a 'social protocol' for the use of each tool at the start of their experience with it.

The four tools used by the students were as follows:

- Email – all students had pre-existing experience of using email for at least their first two years at Durham. Thus email was able to act as a 'control' for experiences with the other tools. A standard Unix mail system is used at Durham, which students access either using pine on the Unix machines, or using a POP client (Netscape, Eudora etc) on a PC.

- BSCW – given the experience both of the students in using this tool, and of the department in supporting it, this was a natural choice. Because of the heavy loading of the BSCW server at Durham by the present second-year using SEGWorld, it was decided that for the CSCW discussions, students would use the publicly accessible server hosted by GMD in Germany.

- TCBWorks – this is a Web-based discussion and voting system, loosely based on the Issue-Based Information System (IBIS) structure. It is hosted by the University of Georgia (US) (Business,).

- CoMentor – a system from the University of Huddersfield (UK) which supports discussion using a MUD (multi-user domain) environment. In the version we used, it is hosted at Huddersfield (Huddersfield)

Towards the end of the course, students were asked to evaluate these four tools based on their experiences. This evaluation used a framework that had been presented on a number of occasions during the lectures as a way of understanding the many perspectives for evaluating CSCW systems. The framework derives from Ramage's PhD thesis (Ramage, 1999), and is described there as a first step towards a heuristic method for CSCW evaluation (Nielsen, 1993). It is presented in two ways, as a set of questions and as a diagram which shows the inter-dependency of the issues involved, specifically that (after the style of the 'systems hierarchies' discussed by Checkland, (Checkland, 1981)) the 'higher' evaluation criteria are dependent for their effectiveness on the 'lower' ones. This is expressed in a series of concentric circles, which gives the framework its informal name of the 'onion model' (figure 6):

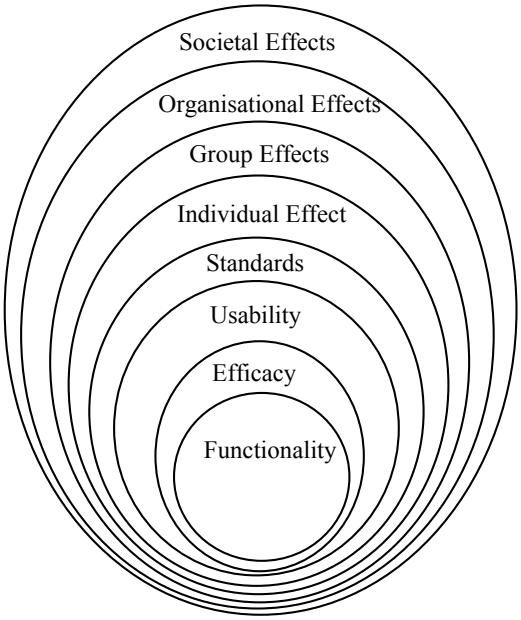

Figure 6: The 'onion' model of evaluation

The eight layers here can be expressed as a set of questions (going from the inner layer out), which can be asked about a CSCW system.

1. Does it work? *(functionality)*
2. Does it work well enough? *(efficacy)*
3. Is it workable with? *(usability)*
4. Does it follow the *standards* laid down by various bodies?
5. What does it do to those who work with it? *(individual effect)*
6. What does it do to their work? *(group effects)*
7. What does it do to those they work with and for? *(organisational effects)*
8. What does it do to the world beyond work? *(societal effects)*

For this exercise, students were presented with a table which showed the four tools they had used as columns, and six of the eight questions above as rows – issues of Standards and Societal Effects were omitted as not being useful on this occasion. Around thirty students were present at the lecture when the exercise was conducted: they were asked to discuss the issues involved in the groups they had divided into for their practical work, and to fill in the table either individually or as a group. Eighteen completed tables were returned. As this number is too small (and the results too varied) for meaningful statistical analysis, we present instead a summary of the comments given for each system (table 1). We will focus here only on the students' evaluations of email & BSCW: the latter can be readily compared with the results from their earlier BSCW experiences, while the former can be regarded as a 'control group', a benchmark groupware system against which all their other groupware experiences were measured. The way the relationship between the two tools was viewed can be summarised by the following comment from one student: "email [is a] great asynchronous tool - invaluable, but other collaboration tools are needed in an organisation - like a shared workspace, for example BSCW".

8. Conclusions

Our Computer Science Department, as an organisation, has benefited from the introduction of groupware into its teaching support systems. Our software engineering groups and their tutors are more effectively supported. There is greater visibility of everyone's work and this makes the process of group working more transparent for both staff and students. As software engineers, we have all gained a first hand appreciation of the impact of introducing groupware into our department. Observing how groups work within their workspaces has given us a better understanding of their work both as a group and as individuals. When students came to use groupware tools in the CSCW module, they did appreciate the need to establish a protocol regarding how they would use a specific tool in order to carry out their discussions effectively. In the SEG work, many students were critical of SEGWorld and their criticisms focused on the technical deficiencies of our BSCW server. The final year students with the benefit of wider groupware experience and more knowledge of CSCW focused more on the outer layers i.e. the non-technical aspects when evaluating the effects of using groupware to organise and co-ordinate their discussions. In some cases, they simply took the functionality for granted.

The group effects were more noticeable within the CSCW student discussion groups. Perhaps this is because the students in the final year of our degree course have more flexibility in their module selections and unlike the second year SEG students are not all taking the same modules and so less likely to be meeting up in classes on a daily basis. Therefore carrying out discussions asynchronously suited the CSCW students who would have found meeting regularly face-to-face difficult. The CSCW students more widespread use of the groupware may also have been influenced by the fact that they knew that they were being marked on the basis of their contribution to discussions, while the SEG students receive no explicit marks for their use of SEGWorld.

The CSCW students' use of BSCW functionality was not significantly greater than their usage during SEG although they clearly understood more of the usefulness of functions particularly those to do with awareness. As a result of the requirements engineering and CSCW lectures on usability, they had a greater appreciation of the BSCW user interface. When working on developing deliverables for the SEG, the main challenges to students are the technical and managerial issues that arise in software engineering projects. In the CSCW module, effective use of the groupware was essential in order to carry out the discussions set. Requiring the students to explicitly develop protocols forced them to consider the relevant functions and their effective use up-front whereas the SEG students only receive a tutorial introduction to SEGWorld and are left to establish their own group's working practices independently.

By its very nature, the SEG project is an exercise in collaboration among the students working in a group. Some of the newly introduced software engineering practical exercises specifically focus on introducing the students to computer-support for collaboration based on tasks which previously were done without the benefit of such support. The CSCW module has offered these same students the opportunity to undertake controlled evaluations of other groupware tools thus exposing them to a different usage, i.e. group discussions. The link between the Requirements Engineering and CSCW parts of the module proved to be fortuitous. Many students were able to bring forward into their CSCW discussions and evaluations principles from the earlier RE lectures.

Any software engineering curriculum must effectively combine theory with practice and must anticipate the world, in which graduates will live and work. The SEG project with its *SEGWorld* environment and the CSCW module both attempt to provide students with group working support, which is realistically close to that which they will find in industry. The CSCW students through their experiences have gained both a theoretical and practical appreciation of how to engineer both social and technical systems that effectively support people in their work.

Through the continued use of these systems within our department we are evolving and extending their application to other areas of our teaching and also to support the work of research groups within the department.

References

Bannon, L., & Schmidt, K. (Eds.). (1991). *CSCW: Four Characters in Search of a Context*: Elsevier Science.

Boldyreff, C. CSCW Module : http://www.dur.ac.uk/~dcs0cb.

Boldyreff, C., Drummond, S., & Walker., R. (1997). Web-Based Support for Software Engineering Group Projects : http://www.dur.ac.uk/~dcs1sad/papers Presented at ECSCW 97, Lancaster University.

Brereton, P., Lees, S., Gumbley, M., Boldyreff, C., Drummond, S., Layzell, P., Macaulay, L., & Young, R. (1998). Distributed Group Working in Software Engineering Education. *Information and Software Technology, 40*, 221 -227.

Business, T. C. o. TabWorks .

Checkland, P. (1981). *Systems Thinking, Systems Practice.* Chichester, UK: John Wiley.

Drummond, S. (1999). *An Investigation into Computer Support for Cooperative Work in Software Engineering Groups.* Unpublished MSc, University of Durham, Durham.

Drummond, S., & Boldyreff, C. (2000). *The Development and Trial of SEGWorld: A Virtual Environment for Software Engineering Student Group Work.* Paper presented at the IEEE Thirteenth Conference on Software Engineering Education and Training, Austin, Texas.

Drummond, S., Boldyreff, C., & Munro, M. (1997). Software Engineering Group Project work: Past, Present and Future, *Presented at Special Interest Group in the Teaching of Software Engineering (SIGToSE), London.* : http://www.dur.ac.uk/~dcs1sad/papers.

GMD-FIT. Basic Support for Cooperative Work (BSCW) : http://bscw.gmd.de/.

Goldberg, A. (1998, 1998). *Building a System in Virtual Reality with LearningWorks.* Paper presented at the Integrating Technology into Computer Science Education, Dublin, Ireland.

Gotterbarn, D., & Riser, R. (1994). *Real-World Software Engineering: A Spiral Approach to a Project-Oriented Course.* Paper presented at the Proc. 7th Conference on Software Engineering, San Antonio, Texas, USA.

Grinter, R. E. (1997). *Doing Software Development: Occasions for Automation and Formalisation.* Paper presented at the Fifth European Conference on Computer Supported Cooperative Work, Lancaster, UK.

Grinter, R. E. (1999). Workflow Systems: Occasions for Success and Failure. *Computer Supported Cooperative Work: TheJournal of Collaborative Computing, 9*(2).

Grudin, J. (1994). Groupware and Social Dynamics: Eight Challenges for Developers. *Communications of the ACM, 37*(1), 93 - 105.

Grudin, J., & Palen, L. (1995). *Why Groupware Succeeds: Discretion or Mandate?* Paper presented at the Proceedings of the Fourth European Conference on Computer-Supported Co-operative Work, ECSCW 95, Stockholm.

Habra, N., & Dubois, E. (1994). *Putting into Practice Advanced Software Engineering Techniques through Students Project.* Paper presented at the 7th SEI Conference on Software Engineering

Education, San Antonio, Texas, USA.

Harrison, J. V. (1997). *Enhancing Software Development Project Courses Via Industry Participation.* Paper presented at the Tenth Conference on Software Engineering Education and Training, Virginia Beach, Virginia, USA.

Huddersfield, U. o. CoMentor : http://comentor.hud.ac.uk/.

Layzell, P., Macaulay, L., Young, R., Boldyreff, C., Drummond, S., Brereton, P., Lees, S., & Gumbley, M. (1998). *Developing a Virtual Community for Student Groupwork: Final Report* : Department of Computation, UMIST.

Mark, G., Fuchs, L., & Sohlenkamp, M. (1997). *Supporting Groupware Conventions through Contextual Awareness.* Paper presented at the Proceedings of the Fifth European Conference on Computer Supported Co-operative Work, ECSCW 97, Lancaster, UK,.

McDermid, J. A., & Rook, P. (1991). Software development process models. In J. A. McDermid (Ed.), *Software Engineer's Reference Book* (pp. 1 -35): Butterworth-Heinemann.

Nielsen, J. (1993). *Usability Engineering*: A P Professional.

Orlikowski, W. (1992). *Learning from Notes: organisational issues in groupware implementation.* Paper presented at the Proceedings of the .Conference on Computer-Supported Cooperative Work (CSCW 92).

Ramage, M. (1999). *The Learning Way: Evaluating Co-operative Systems.* Unpublished PhD, Lancaster, University of Lancaster, UK.

Robillard, P. N. (1998). *Measuring Team Activities in a Process-Oriented Software Engineering Course.* Paper presented at the 11th Conference on Software Engineering Education and Training, Atlanta, Georgia.

Criterion	Email	BSCW
Functionality	Determining the functionality of email in general is difficult. Some students listed functions like sending messages to groups or individuals, sorting messages, filtering, including text and replying. However, many responses seemed to take this for granted and dwelled on the fact that it "lacked a lot of functionality", though this was not specified. The basic view about email was expressed in one response that simply said: "it works, because it has been around for long and everyone knows how to use it".	Most students regarded the functionality of BSCW as very good. One commented that it "works very well for all aspects of groupwork". Capabilities for awareness of others' actions, for different type of documents (and other objects) and different levels of file permissions were noted by several students. One remarked that of the systems they had tried, it was the "most sophisticated for asynchronous work".
Efficacy	Again (as students noted), this depends on the client program used. Speed seemed to be more acceptable than with BSCW, though some had experienced delays using automated mailing lists. Few reported other problems with the effective working of their email tool, although again this may be due to long experience.	In general, the tool seemed to do all that was promised effectively. However, the slow nature of the tool was remarked upon by most people. This made them more likely to make longer points, and in effect made their use of the tool entirely asynchronous (although it can also support synchronous working).
Usability	Most students, given their long experience with email (which some remarked on) found it very usable. Some functions, especially filtering and sorting by subject, were especially noted as increasing its usability; a lack of threading decreased it. Awareness of whether others are online was felt to be lacking, although as most check email very regularly, they were aware of new messages. Automated mailing lists (via majordomo) helped.	Students were very impressed by BSCW's "excellent" UI – it was felt to be "very easy to work with". Some said that it was at the limit of the functions that could be sensibly handled: more would have made it cluttered. One remarked that continually reloading the page of messages to find out if new ones had arrived was a nuisance, although it was useful to get emails to tell them this.
Individual effect	Students felt that email integrates into other working practices, which is efficient, as well as sometimes substituting for the telephone and face-to-face contact. It encourages informal communication, although this can both help and hinder effectiveness! Short messages were felt to be more common than longer ones (this is perhaps partly due to the short time between receipt and reply among the students, who would often use email almost synchronously, as for a conversation).	By contrast with email, longer messages were commoner than short ones. The ease of use of the program was felt by some to encourage "a feeling of professionalism". The awareness function helped to diminish feelings of isolation during discussions. For some, it made them feel "more part of a group at work".
Group effect	Respondents felt both that email "segregates & individualises" and that it "allows personal contact between individuals", the former referring to the potential for conversations to develop among individuals, ignoring the whole group. Again, the difficulty of threaded conversations was mentioned.	BSCW was said to be effective in "maintain[ing] the group, since all members can see all the work". The shared workspace concept was regarded as helpful, as was being able to "organise and document discussions".
Organisational	Email serves to "foster friendship and intimacy among group	BSCW was said to be effective at assisting organisational

effect (mostly from speculation rather than experience)	members". Full mailboxes, especially with (perceived) junk mail or many messages in a discussion, can be a distraction from work.	learning – it "makes the work of the organisation more visible", and helps it to "easily identify group members & contributions".

Table 1: Comparison of Email and BSCW