Open Ended Group Projects, Motivating Students and Preparing them for the 'Real World'

Mats Daniels,
Department of Computer Systems, Uppsala University, Uppsala, Sweden
matsd@docs.uu.se

Xristine Faulkner,
CISE, School of Computing, South Bank University, London UK.
xristine@ntlworld.com

Ian Newman,
Department of Computer Science, Loughborough University, UK.
I.A.Newman@lboro.ac.uk

Abstract

This paper brings together three different sets of experiences of using large scale open ended group projects as a means of enhancing the skills of undergraduate students and preparing them for a career in Software Engineering. The paper commences by examining the circumstances in which open ended group projects have proved to be successful, briefly indicating the criteria that have been used to measure that success. It then examines some of the choices that must be made in designing and managing such projects, drawing on the different experiences of the three authors. It is argued throughout that open ended group projects are an effective way of developing and enhancing skills and of reinforcing learning in most knowledge areas. It is further suggested that open ended group projects offer a way of preparing students for their subsequent careers which is valued by industry and commerce and which also motivates and enthuses students.

1. Introduction

This paper suggests that open-ended group projects (OEGP) are both an effective way of preparing undergraduate software engineers for their future careers and a means of developing and enhancing the skills and knowledge of the students. The paper presents a framework for describing OEGP and uses this to illustrate the wide variety of possible ways in which OEGP can be successfully included in the curriculum by drawing on the experiences of the authors working independently in different institutions [1, 2, 3, 4, 5].

The framework is then used to examine some of the choices that arise in designing and managing OEGP to meet specific educational goals, while also considering how reservations about the use of OEGP in an educational context can be addressed and overcome. In particular, the questions of the choice of suitable tasks, the creation and management of groups and the fair allocation of marks, are all considered.

The paper concludes by stressing that, despite significant differences in the OEGP they have run, the authors all find them effective in generating enthusiasm from the students, in developing the skills and knowledge of the students and in preparing them for work. The

Proceedings of the 15th Conference on Software Engineering Education and Training (CSEET'02)
1093-0175/02 $17.00 © 2002 **IEEE**

authors have also found OEGP to be amongst the most interesting and rewarding of the modules that they have taken, possibly a consequence of student enthusiasm [6].

2. Using Open Ended Group Projects Successfully

2.1 What Constitutes an Open Ended Group Project

It is assumed that the OEGP being discussed forms part of the assessed curriculum for undergraduate students who are expected to either become, or work closely with, software engineers. However, a major difficulty of writing about the experiences of the authors is that the terms for teaching elements associated with higher education vary from place to place, from time to time and from curriculum definition to curriculum definition. A set of terms is, therefore, specified below simply to aid clarity and consistency within the paper.

Programme: The complete planned undergraduate educational experience over three, or more, years, optionally including work experience (internship);

Year: An educational time period corresponding to the progression through the academic process (e.g. first year, second year, final year);

Mester: (a word coined specifically for the paper since 'semester' is used to cover two, three and four periods within a Year in different institutions); a sub period within a Year at the end of which there is an assessment process;

Module: The smallest separable educational unit, typically students will be required to take one or more modules per mester, modules can spread over several mesters though not, usually, over more than one Year;

Module credit: regulations for a degree typically specify the requirements for obtaining the qualification in terms of the number of modular credits needed to pass a year, the credit for a module usually depends on the expected workload and the perceived importance of the module.

Given these definitions, the OEGP which are being considered take place within a module in one or more mesters in a Year of an undergraduate programme.

Within this framework the following 'dimensions' can be used to classify OEGP.

Length: how long do the students have to carry out the OEGP? Mostly this is a mester. Based on the authors' experiences, the shortest allocated period is one week while the longest is a complete academic year.

Simultaneous Activities: what else are the students expected to be doing, as part of their prescribed education?

Scope and Type of Project: what sort of activity are the students being asked to undertake? This can be a research exercise, an actual real project drawn from the outside world or a simulated 'real world' project, but in every case, the characteristic is that the outcome is not predefined.

Cohort composition: what is the academic background of the students involved, are they all following the same programme within the same institution?

Method of selecting and managing the groups: how are the groups formed and what management structure is used?

Modular credit awarded (as a fraction of the credit required to pass the year).

Position within the programme: which year, and within the year, which mester(s).

2. 2 Some Examples of Successful Open Ended Group Projects

This subsection uses five examples to illustrate the breadth of possibilities for OEGP. Each example is explained in terms of the dimensions that were given in the previous sub-section. The details of the projects are kept to a minimum here with a reference being given to the published literature in each case. A discussion of the measures which have been used to judge the examples as successful is included in the next sub-section.

Runestone [5]

A complex problem including aspects of real-time systems, software engineering, computer networks and distributed systems set in an international setting where US and Swedish students work in virtual teams involving both cultural and time zone differences.

Length: 1 mester (10 weeks).

Simultaneous Activities: 2 other modules in the same mester.

Scope and Type of Project: a capstone project giving a simulated real world experience with both a technical and social skill focus.

Cohort composition: the students belong to two different institutions, but are homogenous at either side of the Atlantic.

Group size and composition: most groups consisted of 6 students, three from each country. Last year there were 16 teams altogether.

Method of selecting and managing the group: The groups were put together by the instructors and a fairly strict model of management was used.

Modular credit: The module was worth 7.5% of a whole year.

Position within the programme : The Swedes were all third year engineering students in a 4.5 year programme and the Americans were all final year CS majors in a four year programme.

Brewsers [2]

The project was to look at the effect of feedback on subjects willingness to wait for pages to download from a Website. Each group was asked to build a different interface to the same system. The type of interface was given but not the design for the interface. The build was typical of the type of engineering that students might do in the future.

Length: 1 mester (12 weeks out of a 15 week unit).

Simultaneous Activities: three other modules.

Scope and Type of Project: research project and simulated real world project. The research element was obtained via the evaluation exercise they carried out on their systems. This material was intended to be used by the tutors and the students were made aware of the aims of the project. The students were also

told that the work they did might be passed on to a second group in the following year if the evaluation exercise was inconclusive.

Cohort composition: the students were studying on two different programmes in the same institution.

Group size and composition: The groups consisted of 4 students per group. This was considered by the tutors to be on the small size for each group.

Method of selecting and managing the group: the groups were selected by the tutor and a team leader to head up the build was nominated by the tutor. In addition, each group had to put one person in charge of their evaluation co-ordination.

Modular credit: the unit makes up one eighth of the entire year. The project formed the entire coursework for the unit and consisted of 30% of the final mark .

Position within the program: Final year and second mester within the year (two mesters per year). Most of the students on the unit had previously studied an HCI module with the same tutors . The preceding module combined HCI theory with practical building skills using Java.

The IT Help Desk Support System [4]

The task for each of the groups was to create demonstrations for two complementary web based IT help desk support systems. One would be used by Help Desk staff in a company to record reported problems with IT systems, allocate them to engineers to investigate and publish solutions and work-arounds. The other would be used by employees to check on progress with problems. An integral part of the project was to use prescribed software to produce a dynamic web based project management reporting system which was used for communication with the staff supervising the module.

Length: 1 mester (11 notional weeks, 14 elapsed weeks).

Simultaneous Activities: five other modules.

Scope and Type of Project: simulated real world exercise (the group were told that they were bidding for a contract to produce the actual IT Help Desk support system for a company and that they must negotiate with the staff to agree the scope of their project).

Cohort composition: the students were studying on three different programmes in the same institution but not based in the same department although all of the programmes shared some computing modules.

Group size and composition: the preferred size was 6 people per group, however groups from 5 to 7 were permitted although only groups of six or fewer were eligible for an externally judged prize. In practice, of the 26 groups that were formed, two were of only 4 people, five of 5 and three of 7. Most groups had members drawn from more than one programme - this was strongly encouraged by the specification of the project .

Method of selecting and managing the group: with the exception of one group of four (those people who were not in a group at the end of week 3) the groups were self selected. No management structure was imposed but they were required

to produce and maintain a project management web site to report what they were doing and how tasks were being allocated.

Modular credit: the module was 'worth' one twelfth of the credit for the year.

Position within the programme : Second year, second mester (two mesters per year) reinforcing as much of the conventionally taught material as possible and preparing them prior to going into industry or commerce for an intern year.

Robocup [1]

The ultimate aim is to win the soccer world cup for robots, but the current project 'simply' challenges a large and mixed group of students to build something functioning out of custom made hardware.

Length: the project runs for two mesters for one student group and for one for the other.

Simultaneous Activities: there are no other activities during the mester(s).

Scope and Type of Project: research project and simulated real world project.

Cohort composition: students were studying two different programmes in the same institution, the two mester students were fourth year engineering students and the one mester students were third year engineering students.

Group size and composition: all students formed one team contributing to the achievement of a shared goal. The two mester students were subdivided into sub-groups whose size depended on the task they were in charge of. The one mester students were grouped in teams of 5 or 6 that were assigned to the subgroups. Last year there were 45 two mester and 35 one mester students.

Method of selecting and managing the group: the groups were self selected and they had some freedom to handle the management of their work within a framework.

Modular credit: the module was worth 50% of a whole year for the two mester students and 25% for the one mester students.

Position within the programme: the two mester students were doing the second half of their 4th year in a 4.5 year programme and the one mester students were doing the third mester out of four in their 3rd and final year.

Systems Projects [7]

The groups bid for a project from a list supplied by staff. Each group has a different project but most include a substantial element of software engineering.

Length: 2 mesters - the whole year (21 notional weeks, October-May).

Simultaneous Activities: five other modules in each mester.

Scope and Type of Project: mixed: research project, real world exercise, simulated real world exercise .

Cohort composition: homogeneous, all students in the same year of one programme.

Group size and composition: the recommended size was 5 people per group, however groups from 4 to 6 were permitted.

Method of selecting and managing the group: self selected (except for 5 students out of ~150). No management structure was imposed on the groups.

Modular credit: the module was 'worth' one sixth of the credit for the year as a whole.

Position within the programme : third academic year of a degree programme consisting of four academic years plus an intern year.

2.3. Measures of Success

All of the OEGP described above were successful by internal standards. Within the academic programme the success rates on the module were at least as good as on other modules. However, this could be, in effect, a self-fulfilling prophesy since the staff who were enthusiastic about the idea of OEGP and who set and managed the work also marked it. This sub-section, therefore, concentrates on other, 'external', measures of success.

Firstly, feedback from the students has been generally very positive. In all of the OEGP with which the authors have been involved there has been positive feedback from the students both during the module and afterwards. It is also noticeable that the levels of motivation of the students appeared to be higher with better completion rates, less plagiarism and very few drop outs or failures.

Other measures of success can be found in the readiness of employers to accept students from a programme containing OEGP and in reports from employers on comparative performance of students with and without experience of OEGP during internships and after graduation. There are several instances of this on record with major employers explicitly choosing to try to recruit students from a particular programme because they like the 'product' (graduates or interns). In some cases this has taken the form of offering prizes for the best group performance in an OEGP module, in others the employer restricts their recruitment activity to the output of a few selected programmes and includes ones which contain OEGP. Where OEGP take place before students spend a period in industry or commerce as interns, it is frequently reported that they outperform interns from apparently comparable programmes which do not contain OEGP.

The experience of group project work prepares the students for their subsequent careers where group working is the norm. Undertaking open ended projects also appears to have the benefit that they force the students to think about the problem rather than spending time searching for the 'correct' answer. This has carried over into success in national and international competitions for students who have had experience of OEGP [4, 7].

It is also noted that OEGP appear to have measurable beneficial effects on student performance in other academic subjects. Improved motivation and greater enthusiasm seem to carry over into general performance, confidence levels go up and problem solving skills improve so that students are more willing to attempt difficult tasks. OEGP can also be used to encourage students to apply theory which should lead to a better understanding of the theory and thus to improved performance in examinations [3, 4].

2.4. Educational Aims and Outcomes - Improving Knowledge and Skills

An OEGP is a form of experiential learning [8] which can, in principle, be used to advantage to teach any subject with a practical application. Since software engineering is explicitly concerned with putting theory into more effective practice, it should be possible to use OEGP to advantage with most of the SWEBOK knowledge areas [9]. However, any given OEGP can only have a very limited number of educational goals if these are to be adequately covered in the time available and the performance of the students is to be assessed. It has been suggested [10] that students need to have marks allocated if they are to be expected to consider a subject. Thus, a typical OEGP must concentrate on developing, or reinforcing, one or two specific knowledge areas. As examples, OEGP can be used to motivate students undertaking requirements capture or to help them understand systems analysis, software development lifecycles, specific software design support tools, entity relationship modeling, entity life histories, database design, web site design, or web server programming but not all in one project because sufficient emphasis, in terms of marks, cannot be placed on all of these elements.

In addition to supporting knowledge acquisition, OEGP can be used to help the students gain and improve skills. The most obvious skill areas which are involved are interpersonal communication and group working. However, a suitably designed OEGP can ensure that the students must consider the problems of communication with manager and client and can help improve both report writing and presentation skills. OEGP also assist in getting students to analyse problems and synthesise solutions while examining, and trying to mitigate the risks of things going wrong, all valuable skills for the software engineering project managers of the future [11].

3. (Some of) The Choices in Designing and Managing Open Ended Projects

This section focuses primarily on the choices that the authors have considered when designing their own OEGP which are also expected to be the primary academic considerations for anyone who wishes to include OEGP in the curriculum at their own institution. It commences with a brief discussion of some of the political reservations which have been encountered when the use of OEGP has been proposed and links the choices to be considered with the dimensions given in subsection 2.1.

3.1. Overcoming Political Reservations

The main concerns that are expressed when the use of OEGP are suggested relate either to the use of group projects at all ("weak students get 'carried', good students get 'pulled down'") or to the fact that the outcome for an OEGP is inherently unknown i.e. that there is no 'right' answer. The necessity for group working is, however, becoming more widely accepted now [11], thus it is the concerns about the open ended nature of the project and the need for fair assessments based on problems for which there is a correct answer, which are addressed here. The obvious counter argument is that OEGP mirror real life software engineering projects which do not usually have known 'right' answers and there is a need to assist students to learn this before they start work. Part of this learning process includes

the intrinsically difficult process of finding out what the client thinks is required [12], negotiating with the client to agree what can be done and, later, explaining what has actually been done and how it relates to the requirements. An alternative argument is that, ultimately, all criteria are established and judged by people and are, therefore, subjective. Objective criteria are only regarded as objective because there is agreement about the 'correct' way in which something should be done, or said. History suggests that most such agreements change over time and the current 'right' way may well be revised later.

A different perspective on the fair assessment of OEGP can be provided by considering the way in which science and engineering are advanced. All research projects have unknown outcomes but the methods used to undertake and present research are common. Thus it is possible to provide a fair assessment process for OEGP by focussing on the process which the students use rather than the product which they produce.

3.2. The Choices Inherent in the Dimensions

Each of the dimensions is briefly considered below.

Length: In most cases where the OEGP is to 'fill' a module, the length is a given within the institution. As a general rule, to get the full benefit from OEGP, it needs to be long enough for there to be time for the students to experience difficulties, overcome them, reflect on the process and learn from it, this cannot really happen in a week but can in five or six weeks.

Simultaneous Activities: the pattern of modules is usually a given within the institution. Experience appears to show that there is little difference in the learning outcomes provided that the elapsed time is realistic. One possible worry is that the increased motivation for students undertaking OEGP could lead to them spending a disproportionate amount of time on the OEGP module to the detriment of other modules taking place at the same time. However, this has not been the experience of the authors. Provided the students are reminded that the other modules are also important, there is usually no overall problem and performance in other modules can even improve [3].

Scope and Type of Project: this is considered in more detail in 3.3. The experience here is that the project has to be seen by the students as large enough to be a challenge but not so large as to be too daunting to start. One major factor which affects the students' perception is the enthusiasm and supportiveness of the staff. Students will achieve quite remarkable results if the staff convince them that they can.

Cohort composition: for staff and students the most satisfying OEGPs, and the nearest to a realistic experience, occur where there is a 'mixed' cohort and the students are in groups involving people with different backgrounds.

Group size and composition: given a desire to improve intra-group communication skills, group management and work allocation skills and to give them the opportunity of experiencing intra-group conflict, and resolving it, the recommended size of a group is 'about' 6. Groups of 2 or 3 have fewer

communication or conflict 'opportunities' and groups larger than 7 seem naturally to subdivide and work as smaller groups.

Method of selecting and managing groups: discussed in 3.4.

Module credit: where OEGP 'fill' a module, the modular credit is a given. However, it is possible to have some modules which involve more work and carry more credit. As noted in 2.1, the experience of the authors includes credits ranging from one twentyfourth to one half of the Year's assessment. The key factors seem to be that the students perceive that the amount of credit is 'worth it' and 'fair'. A more interesting topic, something that the students feel they must find out about , may offset a relatively small amount of credit.

3.3. How Open Ended and How Large - Setting Goals and Formulating Assessment Criteria

This sub-section is concerned with the choices relating to the task itself, the educational objectives and matching the assessment to the objectives and the task.

Choice of Task

The task selected must be something that will interest and challenge the students and which is of interest to the staff. The viability of different task types is considered first and then the 'size' of the problem.

Real world problems with real clients seem, initially, to be ideal. The real clients can explain the problem and interact with the students. However, this ignores the educational issues, which may also result in very dissatisfied clients. The client has a problem and would like a solution. The students are undertaking the task of solving the problem, not primarily to produce a solution but to learn how to solve problems like this and to gain academic credit for their attempts. The educational objectives are about gaining knowledge and skills and the students may learn as much, or more, from making mistakes as from 'getting it right'. The client needs to understand, from the outset, that the most they can realistically hope for will be that they have explored the problem more thoroughly. Simulated real world problems where the academic staff are the clients overcome the potential difficulties associated with having real clients. Research projects can be an excellent way of offering the benefits of having real clients who have realistic expectations of the likely outcomes. However, it is still necessary to recognise that the educational outcomes must take precedence for the students over the research objectives.

Viable problems can have a significant variability in size since 'size' itself has several dimensions. The first distinction that needs to be made is between product and process .

If the main assessment criteria are to do with the process that the students follow rather than with the product they produce, then it is not necessary for them to produce a complete product in order to be successful. If they can explain what they did do and also explain what they should have done and why, then they have learned something useful from the OEGP, even if there is no viable product. Analysing the problem, agreeing requirements with the client, agreeing assessment criteria for the product, producing partial designs for a product, possibly including prototypes, would all be worthwhile steps in the development of a professional software engineer without any actual product being produced. The final

product would also not necessarily be important if the objectives were to give the students experience of group working and conflict resolution or if the OEGP were being used as a vehicle to give the students practical experience of, and skills in, the use of a design or development support tool (e.g. Rational Rose). In all these cases the primary objective is that the task should offer requisite variety so that it poses challenges which will test and develop the relevant knowledge and skills of the students.

Educational Objectives and Assessment

OEGP can be used to develop knowledge (e.g. database design principles), software engineering related skills (e.g. use of a design support tool), or personal skills (e.g. time management or communication). Most OEGP are designed to help the students gain a combination of knowledge and skills. It is generally accepted that skills can only be gained through practical work and that group working skills require group work, so OEGP are ideal for these. Communication skills and analysis and synthesis skills also benefit from working within a group of peers. However, OEGP is less useful for developing and reinforcing some of the individual skills (e.g. keyboarding) since there is a tendency for the group to 'specialise' within the task and the person who is most skilled, or most willing is likely to do the majority of this work.

As has already been discussed it is possible to use OEGP to assist students to learn in most knowledge areas. However, if marks are an important motivator, then it is necessary to decide which knowledge area(s) is (are) to be concentrated on in a particular OEGP.

A group project does not inherently lend itself to ensuring that all members of the group gain the same skills or knowledge. However, careful choice of assessment criteria can ensure that all of the students in the group are focussed on the intended learning outcomes and can also ensure that every member of the group has to reflect on what they have learned in the relevant areas. As an example, if time management and group organisation were chosen as important skills, then the groups could be asked to report on a regular basis how tasks were being allocated, and performed, within the group, with each individual being asked to produce a report at the end commenting on how the group functioned with respect to these skills. Similarly, if entity relationship modeling were chosen as an important knowledge area which needed reinforcement, the groups could be asked to submit an ER model at some intermediate point in the project, receive feedback and then be asked to submit a revised report later [4].

3.4 Forming and Managing Groups

This sub-section concentrates on the ways in which groups can be chosen and the ways in which they are managed and are expected to report.

Forming groups

The two basic ways of forming groups are to have them selected by the staff or to allow the students to select their own groups (in which case there must be some provision for ensuring that students who do not join a group are still put in a group). There are advantages and disadvantages with each possibility. If the groups are 'forced' it is possible to form groups of a desired size and, to some, extent to select preferred skill and ability 'mixes', although, this may not be possible, in practice. Pre-selecting the groups also

corresponds to the normal arrangement in industry and commerce where one cannot usually select ones team mates. However, allowing the students to select their own groups transfers some of the responsibility for the process to the students which, in turn, increases the pressure on them to ensure that the group is successful (a good motivating factor). When a group experiences problems because one of its members is not contributing satisfactorily or because of personality conflicts, the forced groups tend to turn to the 'boss', the staff member who selected them, to find a solution. Self-selected groups may ask the boss but can legitimately be told that it is their problem and can be asked to think through their options and then to present the staff member with the options and their recommended decision.

Managing the groups

Two questions are considered here. The first is the method used to get the group to manage itself, organise and allocate work and finish the task. The second is the way in which the staff monitor the progress of the groups to ensure that educational objectives are met and that the groups get the maximum benefit from the experience.

It is possible to prescribe a management style on the groups. They can, for example, be asked to elect a manager and report it to the supervisor and to use particular management styles and reporting techniques. This is particularly appropriate where they have been taught a specific method and one of the educational outcomes is intended to be the re-enforcement of the method. It is also appropriate where the task is deemed to be a difficult one and it is felt that there is insufficient time available to allow the groups to experiment with different management styles in order to find one which is appropriate. This is very likely to be the case with forced groups since they have, in any case, to learn about their individual strengths and weaknesses and how these can best be utilised/overcome. It is also possible to suggest alternative management styles but let the group decide the style they use for themselves. This is applicable where there is plenty of time for experimentation and/or the students already know one another and can quickly establish a modus-operandi. The experience of the authors is that both of these methods work in practice and the choice between them is mainly a matter of personal preference.

Monitoring progress: all the authors believe that it is necessary for staff members to have fairly close contact with the groups in order to monitor their progress and detect groups who are having problems or challenge groups who find the task too easy. However, there are various methods by which this can be achieved. These can vary from personal contact with groups on a regular basis to an e-mail or web based reporting system. Formal group presentations at which all of the group must be present and where the staff members give feedback are a good way of getting the group to concentrate on providing deliverables, they also improve confidence and presentation skills.

4. Conclusions

A case has been offered by the authors for the use of OEGP to support both knowledge and skill acquisition for software engineers. The dimensions of OEGP have been explored and some of the choices associated with the use of OEGP have been analysed. In the experience of all of the authors, OEGP are a very effective way of developing skills and knowledge. Students who have had experience of OEGP in academia are better prepared

for a subsequent career and the experience is valued by employers. However, OEGP do make different demands on staff when compared to conventional taught modules. Since there is no right answer, it is necessary to consider suggestions from students carefully and to try to remain open minded and flexible in discussions with the students. However, if the staff can take this approach, OEGP seem to offer many advantages with few drawbacks.

Two specific advantages which might not be anticipated when OEGP are suggested for introduction into the curriculum are a substantial improvement in student motivation and an absence of plagiarism. The students tackle the work enthusiastically, put in additional work and interact with staff more often and more purposefully than for other modules. It is noticeable that the completion rates are higher than for conventional modules and the drop out rates are lower. It is also observable that the average standard goes up and more students pass OEGP modules than conventional modules. One explanation for the reduction in plagiarism is, probably the improved motivation. Another possibility is the closer contact with staff which makes it seem to them more likely they will be detected. However, the most probable reason is that the open ended nature of OEGP, the absence of a 'right answer' and the increased interest that this generates in the students, make it easier for the students to generate an answer of their own than to try to adapt an answer produced by someone else to their particular circumstances.

5. References

[1] M. Daniels, L. Asplund, "Multi-Level Project Work; A Study in Collaboration", 30th ASEE/IEEE Frontiers in Education Conference", 2000.

[2] X. Faulkner, F. Culwin, "From Tuttles to Brewsers: Integrating HCI and Software Engineering Using the Whole Class Project", Proceedings of Project 99, 1999.

[3] D. Parish, I. Newman, "Educating Systems Engineers in the University sector", IEE Engineering Science Education Journal 8, 1999, pp169-175.

[4] I. Newman, R. Dawson, L. Parks, "Reflecting on the Process, Some Experiences of Teaching Students to Think About how they Produce Software in a Real Environment", in INSPIRE V Quality & Software Development Teaching and Training Issues, British Computer Society, GB, 2000, pp 25-36.

[5] A. Pears, M. Daniels, A. Berglund, C. Erickson, "Student Evaluation in an International Collaborative Project Course", Proceedings 2001 Symposium on Applications and the Internet Workshops, IEEE Computer Society, USA, 2001, pp74-79.

[6] D. Beckett, "Disembodied Learning: How Flexible Delivery Shoots Higher Education in the Foot, Well Sort of.", Electronic Journal of Sociology: 3, 3, 1998, http://www.icaap.org/iuicode?100.3.3.2

[7] I. Newman, "Observations on the Relationships between Initial Professional Education for Software Engineering and Systems Engineering - a Case Study", Proceedings of the 14th Conference on Software Engineering Education and Training, IEEE Computer Society, USA, 2001, pp172-181.

[8] Kolb, D., Experiential Learning - Experience as the source of learning and development, Prentice Hall, New York 1984.

[9] "The Guide to the Software Engineering Body of Knowledge" (www.swebok.org) Stoneman Version 0.95, April 2001.

[10] B. Friedman, P. Kahn, "Educating Computer Scientists", CACM 37, 1994, pp65-70.

[11] G. Ford, N. Gibbs, "A Mature Profession of Software Engineering", CMU/SEI-96-TR-004, January 1996.

[12] R. Veryard, "Demanding solutions", Requirements Engineering Vol1 No4 1996 pp261-263.