London Metropolitan University

TOWARDS UNDERSTANDING AND IMPROVING THE PROCESS OF SMALL GROUP COLLABORATIVE LEARNING IN SOFTWARE ENGINEERING EDUCATION

by

Peter Kehinde Oriogun

A thesis submitted to
LONDON METROPOLITAN UNIVERSITY
for the degree of
DOCTOR OF PHILOSOPHY
PhD by Prior Output

May 2006

DEDICATION

I am dedicating my PhD by Prior Output to my entire family. Firstly my mother, Mrs. Phebean Ebunoluwa Oriogun, my late father, Mr. Emmanuel Oluwakayode Oriogun. I thank you both for your guidance, tolerance, perseverance and love throughout the years.

To my brothers and sisters, Mr. Paul Taiwo Oriogun, Mrs. Victoria Idowu Oriogun-Williams, Mrs. Dele Omowumi Oriogun-Van Der Ligen, and Mr. William Akinwale Oriogun. I am very proud to have you as my brothers and sisters, and I thank you for understanding the importance of having a strong family unit. I love you all.

To my children: Chantelle Oluwatosin, Titilayo, Oluwakemi, Temitayo and Oluwashola. You are all my source of inspiration and my reason for being. My love for each of you is more than words can say.

Finally to my nieces and nephews: Remi, Femi, Yemi, Tola, Lucas, Luyi and Loni. I thank you all with much love for being members of my family.

May Christ in His infinite wisdom and mercy, continue to bless and guide each and every one of us today and always. Amen.

ACKNOWLEDGEMENTS

I would like to acknowledge a number of individuals for their contribution to my research activities over the past 8 years. Special thanks goes to Professor Yau Jim Yip, who encouraged me to develop my personal research.

A big 'thank you' goes to my good friend, colleague, co-author and co-Director of Study, Dr John Cook. A sincere 'thank you' goes to Professor Tom Boyle, my Director of Study, whose encouragement and professional approach to supervising the authoring of this report has been of immense help. A special 'thank you' goes to members of the Learning Technology Research Institute, at the London Metropolitan University. A 'thank you' goes to Colin Rainey, who proof-read this thesis.

Special thanks also goes to Professor Robert Gilchrist, who worked with me on the three-year study on the impact of appropriate prior study for software engineering students. His statistical input and guidance has informed a number of my findings and recommendations in the portfolio of work that I am submitting.

I would also like to take this opportunity to thank my colleagues who have co-authored conference papers and journal articles with me, especially those who have contributed towards the portfolio of work that I am submitting. A special 'thank you' goes to all my undergraduate and postgraduate software engineering students from 1998-2006. In particular those students, who contributed to case studies that form part of my submission.

ABSTRACT

The research aim of this submission for PhD by Prior Output is to understand and improve the process of small group collaborative learning in software engineering education. The research portfolio supporting the submission specifically deals with a number of background studies (the establishment of an optimal software life cycle process model for teaching software engineering in the small group collaborative setting) leading to the development of an appropriate pedagogical approach for underpinning small group learning, understanding the type of learning interaction that was taking place within such small group learning, and finally, the development of appropriate methods for analysing collaborative small group learning in software engineering education.

In the portfolio of work submitted for the PhD, I have systematically investigated my research aim and problem in studies involving 241 different students over a period of 8 years. I contend in my submission that I have made a significant contribution to knowledge in my quest to understand and improve the process of small group collaborative learning in software engineering education within higher education, in order to prepare students for employment in software engineering by (i) developing and testing a documentation toolkit for collaborative problem-based learning (ii) a methodological tool for analysing and understanding inter-rater reliability (iii) a framework for the development of teamwork and cognitive reasoning when learning in small groups.

1.0 Introduction

1.1 Motivation

Educating software engineers is fundamentally based on problem solving through which students assimilate and apply knowledge and skills to problems of varying complexity, size and from diverse domains. The level of understanding of the underpinning theory and acquired skills need to be ascertained through assessment. Traditional unseen examinations have long ceased to be recognised as the sole method of assessment. Furthermore, in recent years we have seen the widespread adaptation of computer mediated communication (CMC) in education, including the extensive interest in using online communications to facilitate asynchronous dialogues, such as online teamwork.

Software engineering tasks typically involve teamwork supported by networked computers. Collaborative group working of software engineering teams is increasingly evident in the 'real-world'. Tools to support such collaborative group work within an education setting are at present limited to general purpose groupware involving video, audio, chat, whiteboards and shared workspaces. Within software engineering education, collaborative group tasks have an established role in the curriculum. However, in general, such collaborative groups are local to a particular university or institution and are normally composed of students who have significant shared history in terms of technical background and social interaction, and able to meet face-to-face on a regular basis.

Having worked in industry as an Analyst Programmer for a number of years before progressing into academia over 10 years ago, working as a Senior Lecturer in Software Engineering at two different Higher Education Institutions in the United Kingdom, I was very much aware of a number of problems relating to software engineering education including the ones mentioned above. The type of software engineering students we attract in UK institutions normally do not have significant shared history in terms of technical background and social interaction, and are not normally able to meet face-to-face on a regular basis when assigned a software engineering problem that involves collaborative effort in determining or finding a collective solution to the given problem.

The main reason for software engineering students within higher education institutions not being able to meet face-to-face is partly due to the fact that the majority of them are non-traditional students; they are predominantly mature students with family responsibilities and with part-time work commitments in most cases. The majority of the younger students have to balance their education with, in some cases full time employment, as they are always worried about paying off the government loans that fund their education at the commencement of their university education. Furthermore, software engineering education has not made effective use of technology in supporting and facilitating collaborative learning for our non-traditional software engineering students. Consequently, non-traditional software engineering students within higher education institutions are ill prepared to tackle real software engineering problems collaboratively and collectively in industry after their university education. This problem motivated me in wanting to understand and improve the process of small group collaborative learning in software engineering education, in order to prepare students for employment in software engineering.

1.2 Research Aim and Problem

The aim of the my research was to understand and improve the process of small group collaborative learning in software engineering education, in order to prepare students for employment in software engineering. An initial literature review revealed that there is strong evidence that software engineering students benefit from and enjoy working in small groups because it gives them a sense of belonging to a community (Habra & Dubois, 1994; Gotterbarn & Riser, 1994; Harrison, 1997; Brereton et al. 1998; Robillard, 1998). However, this initial literature review also revealed that there are no models specifically developed for supporting software engineering in education. This problem was compounded by the fact that software engineering models used in industry were, when adopted to an educational setting, found to be too document heavy and were unwieldy in that students did not get enough chances to solve problems within the time frame of a single semester. More general work in areas such as problem-based learning (Bridges, 1992; Boud and Feletti, 1996; Trop and Sage, 1998) were found useful, but were used mainly in medicine, nursing education and business studies but there was nothing in software engineering. Consequently, my research aim was sub divided into three specific objectives to address this problem as follows:

1. To develop an appropriate pedagogical approach for underpinning small group learning;
2. To understand the type of learning interaction that was taking place;
3. To develop appropriate methods for analysing collaborative small group learning in software engineering education online and offline.

1.3 Outline of the Thesis

The rest of the thesis-supporting document is organised as follows. Section 2 gives the background study to my research objectives, which led to my survey of Boehm's work on the Spiral models, a cross-sectional study on the establishment of an optimal software lifecycle process model for teaching software engineering in the small group collaborative setting online and offline, and my generic architecture supporting software negotiations and reconciliations online and offline. In Section 2, in the case studies presented, there were three variables being studied, namely,

- The software life cycle process models that the students were using;
- The setting of students into groups;
- The prerequisites that students had studied.

Section 3 will explain my reasons for developing a novel Toolkit for scaffolding small group problem-based learning as an appropriate pedagogical approach for underpinning small group learning. Section 4 briefly documents why, after addressing my first two objectives above, it was necessary for me to understand in more depth the type of interaction that was taking place with software engineering students. In Section 5, the findings from section 4 ultimately lead to my major contribution, the development of the theoretical basis for my semi-structured method for analysing small group collaborative learning in software engineering education and a prototype software tool supporting this approach. My semi-structured method for analysing small group collaborative learning is within the generic architecture framework that I proposed, and supports Computer Mediated Communication. In Section 5A, I validated (with colleagues from the learning Technology Research Institute at London Metropolitan University) the cognitive engagement of my SQUAD approach with two established analytical tools for measuring online discourse. Section 6 deals with an experiment suggested by the two external examiners, to validate my TRCP method with existing software engineering students, using students' data from my SQUAD approach. Section 7, will concentrate on my contribution to knowledge. Section 8 will suggest ongoing and future work. Section 9 lists references for PhD thesis by Prior Output covering documentation. Section 10 lists the remainder of my research output from 1999–2006. Appendix 1 contains the hard copies of co-authors' statements. Appendix 2 are the hardcopies of my 11 submissions for the PhD by Prior Output. The timeline table below shows the organisation of this thesis.

Research Objective	Study	Research Method	No of students	Study No	Contribution
	Establishing an optimal software life cycle process model for teaching software engineering in small group collaborative setting.	Survey	1999 None	1	Background study to my research objectives (Survey of Boehm's Work on the Spiral Models).
		Regression Analysis / Case Study	1998 – 1999 62 BSc students	2	Background study to my research objectives (recommended a hybrid between the Win-Win Spiral and Incremental Development would best enhance students' performance in software engineering).
		Regression Analysis / Case Study	1999 – 2000 55 BSc students	3	
		Regression Analysis / Case Study	2000 – 2001 65 BSc students	4	
			2001 – 2002 6 MSc	5	Background study to my

			students		research objectives (Developed my negotiated incremental architecture).
1	Developed an appropriate pedagogical approach for underpinning small group collaborative learning	Case Study Case Study	December 2002 6 MSc students 4 BSc students December 2003 6 MSc students	6 7	Developed my enhanced problem-based learning grid in order to facilitate the development of new courses that included problem-based learning as part of their pedagogical model.
2	Understanding the type of learning that was taking place online with software engineering students in small collaborative learning	Content Analysis / Grounded Theory / Case Study Content Analysis / Grounded Theory / Case Study	April 2003 7 BSc students December 2003 6 MSc students	8 9	Developed my Transcript Reliability Cleaning Percentage (TRCP) for understanding the type of learning interaction that was taking place online with software engineering students in small collaborative setting. Developed the theoretical underpinning of my inter-rater reliability measure of online transcripts (TRCP)
3	To develop appropriate methods for analysing small group collaborative	Case Study / Content Analysis	December 2003 6 MSc students	10	Developed the SQUAD approach, a generic method for analysing and scaffolding small

	learning in software engineering education online and offline				group collaborative learning online and offline.
			December 2005 13 MSc students.	11 New published study.	Validated the SQUAD approach with a rival Canadian method, and an American rival method.
			February 2006 5 BSc Students	12 (Experiment suggested by the external examiners at my viva voce examination) New unpublished study.	Using the SQUAD approach to provide usable assessment results with the minimum staff effort.
			TOTAL Number of Students 241	**TOTAL Number of Studies 12**	A software tool has been developed to realise the SQUAD approach, with the intention of developing the tool to a marketable standard for use within HE institutions

Table 1A: Timeline Table of Studies and Contributions

2.0 Background Study for my Research Objectives

2.1 Software Engineering Education Working Group

A group of industry and academic professionals interested in promoting the development and future outlook of software engineering education and training met in Pittsburgh USA in November 1997 to discuss the urgent need for a set of guidelines to support the design and implementation of software engineering courses and curricula. The group was called the Working Group on Software Engineering Education and Training (WGSEET). They meet twice a year to promote the advancement of software engineering education. They agreed that there existed some early work to support software engineering curriculum development (BCS, 1989; FORD, 1990), however, there was no document that provides broad and comprehensive information and direction for the development of undergraduate programmes in software engineering.

In October 1999, the WGSEET produced a report, entitled 'Guidelines for Software Engineering Education Version 1.0' (Bagert et al., 1999). The report prescribed in detail the topics to be covered within a software engineering curriculum, such as a software engineering body of knowledge, a software engineering curriculum model, curriculum support and curriculum assessment and accreditation. They concluded that software engineering is a mature discipline that is becoming increasing critical in all aspects of human endeavour. They further argued that the demand for well-educated software engineers is increasing, but sufficient computing courses to support this demand do not exist.

The WGSEET report however, failed to mention that software engineering education is a collaborative team effort, and that a framework is required for such a collaborative team for the development of software artefacts making up the final software product being built by software engineering students. Section 2.2 addresses this particular omission from the WGSEET report.

2.2 Towards establishing an optimal software life cycle process model for teaching software engineering in a small group collaborative setting online and offline

From my initial literature review, the software engineering process models used in industry were found to be too complicated and not pedagogically effective in small group learning at the modular level within a higher education setting. Consequently, I conducted a literature review of four iterative software life cycle process models (models used to produce software prototypes) in order to establish the most suitable one to adopt in order to teach small group collaborative learning in software engineering education. The four models were the Rapid Application Development or RAD (Martin, 1991; Keer and Hunter, 1994; Pressman 2000), the Spiral model (Boehm, 1996), Incremental Development method (Pressman, 2000) and the Win-Win Spiral model (Boehm, 1998). The reason for choosing these four particular models was because they are the four main iterative software process models found in software engineering core texts.

My literature review of four iterative software life cycle process models also led to my Study 1, where I conducted a survey of Boehm's work on the Spiral models **(Oriogun, 1999)**, which helped to focus a three year study – Studies 2-4 **(Oriogun, 2001; Oriogun and Gilchrist, 2002)** on the performance of software engineering students in order to establish the appropriate prior study for software engineering education and to compare and contrast the performance of software engineering students using the four iterative software life cycle process models. The research methods used for this study were case study and statistical method (regression analysis of student performance). This study involved profiling students, allocating them into groups, and gathering data on their performance.

2.3 The three-year cross-sectional studies on software engineering students' performance and establishing an optimal software life cycle model (1998-2001)

From 1998 until 2001, I conducted a three year cross sectional study on the performance of software engineering students (at the University of North London, now known as London Metropolitan University after merging with London

Guildhall University in 2002), in order to establish the appropriate level of prior study for software engineering education and to compare and contrast the performance of software engineering students using the four iterative software life cycle process models mentioned above **(Oriogun, 2001; Oriogun and Gilchrist, 2002)**.

In the academic year 1998-1999, from a population of 88 software engineering student, 62 students participated in Study 2. In the academic year 1999-2000, from a student population of 81 software engineering students, 55 students participated in Study 2, and, in the academic year 2000-2001, from a population of 122 software engineering students, 65 students participated in Study 3. Over the 3 years, from a total population of 291 software engineering students, 182 students participated in the study **(Oriogun, 2001; Oriogun and Gilchrist, 2002)**.

For Study 1 in 1998-1999, with a sample size of 62, students were all asked to adopt the classic life cycle software process model, commonly known as the Waterfall model (Royce, 1970) to develop the practical aspect of their software engineering module in a small groups collaborative setting (between 4-6 students per group). This was to serve as a baseline for the 3 year study. The reason for this was to conduct an in-depth analysis on four different prototyping software life cycle process models with software engineering students during the academic year 1999-2000. The Waterfall is a linear sequential software process model, with specific deliverables and milestones over a timeline.

2.3.1 Statistical models of software engineering students' performance 1999-2000

A detailed analysis of the statistics during the academic year 1999/2000 –Study 3, revealed that prior knowledge of an introductory programming module (a year 1 semester 1 module) does not have any effect on students' performance as a body of prior study for software engineering students, it was therefore not significant, and was removed from the two models describing students performance below. The module Information Systems Analysis and Design (ISAD) was found to be statistically significant as a source of prior study for software engineering students in order to enhance their performance. Two forms of statistical models were considered namely, a variate-factor interaction model and a simpler model which forces influential data points to be graphed with the same common slopes, and different intercepts.

2.3.1.1 Interaction model

SE Performance	= 40.87	+0.3204 *ISAD	if **WinWin/ Incremental**
	= 40.87 - 7.6	+0.218 *ISAD	if **RAD**
	= 40.87 -32.6	+0.810 *ISAD	if **Spiral**

(**SE Performance** is the overall performance of software engineering students on the practical small group collaborative aspect of the module –the group coursework).

From the data (**Oriogun, 2001; Oriogun and Gilchrist, 2002**), there was no strong evidence (p=0.06) that the above more complicated (interaction) model significantly explains more of the variation in the data. We therefore preferred the simpler model below without the variate-factor interaction (**Oriogun, 2001; Oriogun and Gilchrist, 2002**). The 3 equations below showing **SE Performance** represent the graph in Figure 1.

The statistical models in Figure 1 (the analysis of the 1999/2000 data) suggests that the expected coursework performance for a software engineering student using the Win-Win Spiral model or the Incremental Development method would be in the range (40% - 70%), although any individual student's performance would vary within the expected range. The second statistical model suggests that the performance for a software engineering student using the RAD approach would be in the range (26% - 60%), although any individual student performance would vary within the expected range, and the third statistical model above, suggests that the expected performance for a software engineering student using the Spiral model would be in the range (29% - 62%), although again any individual performances would vary also.

2.3.1.2 Non-interaction model

SE Performance = 39.94 + 0.335 *ISAD if WinWin / Incremental

= 39.94 -13.86 + 0.335 *ISAD if RAD

= 39.94 -11.14 + 0.335 *ISAD if Spiral

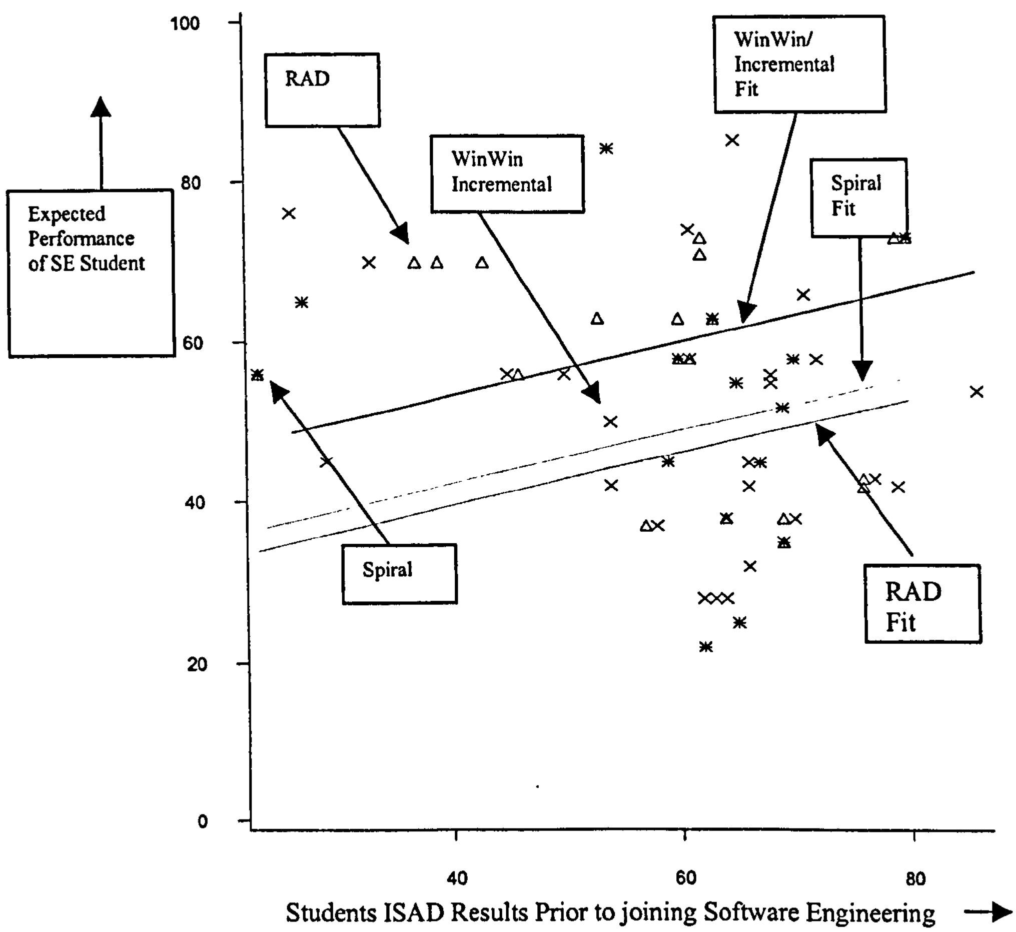


Figure1: Preferred simpler non-interaction model supporting iterative software life cycle process models **(Oriogun, 2001; Oriogun, 2002)**

During 2000-2001 academic year –Study 3, the results of the 1999-2000 Study 2 were revealed to software engineering students at the start of the semester, and they were asked to choose the model they would prefer to adopt for the practical aspect of the software engineering module, in a small group collaborative setting in which to develop their software projects. All of the groups formed (from a total of 65 students) opted to combine aspects of the Win-Win Spiral model with the

Incremental Development method as suggested from the study conducted in 1999-2000 **(Oriogun, 2001; Oriogun, 2002)**.

On average those students who opted to combine the Win-Win Spiral model with the Incremental Development method scored 10% more than those who opted for either the RAD or the original Spiral Model. Consequently, the 3 year studies revealed that the Win-Win Spiral model (1993; Boehm, 1996; Boehm, 1998) was the most effective in supported small group collaborative learning in software engineering education. The 'Win-Win spiral model' achieves this by having a number of stakeholders negotiating, reconciling and resolving controversial and non-controversial software requirements in a systematic and controlled environment. The major findings from the study were that an integrated model, a combination of the Win-Win Spiral model and the Incremental Development method, was the optimal iterative software life cycle process model for teaching software engineering in a small collaborative setting online and offline. The findings have been reported in two of my research papers **(Oriogun, 2001; Oriogun and Gilchrist, 2002)**.

2.4 A generic framework for negotiating software requirements in a small group collaborative setting online and offline

The findings from the three year studies also led to the development of the 'negotiated incremental architecture', a structured framework for managing the collaborative negotiation and construction of software **(Oriogun, 2002)** – Study 5. The negotiated incremental architecture framework is where software engineering students (referred to as stakeholders – see Figure 2) can negotiate software requirements online and offline by providing a generic architecture with a set of procedures or protocols for doing things. This generic architecture later became the basis of the architecture supporting my main contribution to knowledge within my submission **(Oriogun, 2004)**. Figure 2 shows the negotiated incremental architecture. Full detail of its use has been reported **(Oriogun, 2002)**.

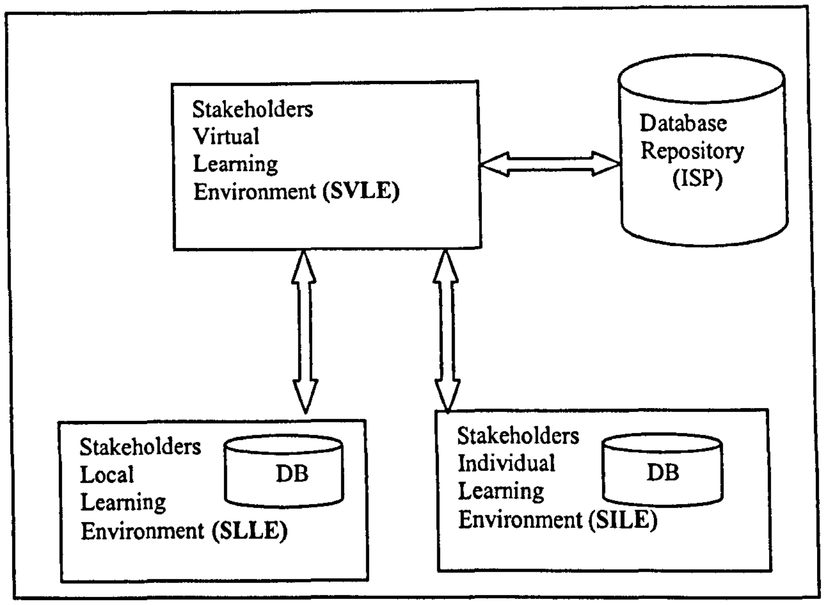


Figure 2. The Negotiated Incremental Architecture –NIA (**Oriogun, 2002**)

During the academic year 1999-2000 –Study 3, software engineering students were asked to join an Internet Service Provider (ISP) in order to facilitate negotiation and reconciliation of software requirements. As a result of setting up a negotiation facility away from the university, students could continue their online collaborative software negotiation and reconciliation when the University server failed. The database repository linked to the SVLE represents the fact that other users may have their information stored with the same ISP. The module tutor is an essential stakeholder, as it is necessary for the student stakeholders to seek clarification in parts throughout the negotiation phase of the software development process. The student stakeholders are encouraged throughout the negotiation phase to play active roles, and rotate the management of the group on a two weekly basis in order for every member of the group to have ownership of the developing software (**Oriogun, 2002**).

The Stakeholder Local Learning Environment (SLLE) is where the students come together as a group at the University to have a face-to-face meeting, about the requirements, as well as recording it online for future reference. At this stage, there may be one member of the group appointed to record the proceedings of the meeting and post it at a later date or immediately onto the SVLE database. The module tutor (who is also one of the stakeholders) could be part of the SLLE negotiation phase otherwise the tutor could access the student stakeholders as within his/her own SILE via the SVLE set up by the group **(Oriogun, 2002)**

The Stakeholder Individual Learning Environment (SILE) depicts the negotiation carried out by each stakeholder away from the university, either by using their own home computer(s) or using other computer(s) to access the SVLE in order to obtain updated information as to the negotiation activities thus far within the group of stakeholders. The SILE is primarily via a computer, however, it is possible that students could access the SVLE via a mobile telephone as well as other telephone lines **(Oriogun, 2002)**.

2.5 Lessons learnt and how the background study to my research objectives resulted in SQUAD

In my research context, it is not possible to understand the full implications of my major contribution, SQUAD without appreciating the type of prerequisites that will benefit software engineering students in order to engage within small group collaborative learning. The lessons learnt from my background study to my research objectives were that an integrated model, a combination of the Win-Win Spiral and the Incremental Development method, was the optimal iterative software life cycle process model for teaching software engineering in small group collaborative settings online and offline (Oriogun 2001: pp108-109). Furthermore, a student's performance in advanced computing modules would be better enhanced if they had studied appropriate modules dealing with the basic concepts and groundwork for the advanced computing module (Oriogun and Gilchrist 2002; pp108-109). The background study to my research objectives resulted in my Negotiated Incremental Architecture (Oriogun 2002), which became the framework for my major contribution, the SQUAD approach.

3.0 Developing an appropriate pedagogical approach for underpinning small group learning

3.1 Investigating educational small group collaborative learning

In parallel with the Studies 2, 3, 4 and 5 above, and in order to develop an appropriate pedagogical approach for underpinning small group learning, I investigated educational small group learning. Specifically problem-based learning (Bridges, 1992; Boud and Feletti, 1996; Trop and Sage, 1998) was found to be most amenable to small group collaborative learning in software engineering education. Problem-based learning has been around since 1960s (Neufeld and Barrow, 1974).

There are no shortages of models and theories of learning. One could easily argue that the plethora of perspectives and technology confuse rather than aid educational planning and, in particular, teaching. Learning is not only complex; it is also a multi-faceted process. It was noted by Kolb (1995) that:

> *To learn is not the special province of a single specialised realm of human functioning such as cognition or perception. It involves the integrated functioning of the total organism – thinking, feeling, perceiving, and behaving. (p. 148)*

Learners are influenced by a myriad of psychological, social and situational factors that pervade their everyday lives. It is possible that as educators, our best efforts in trying to educate our students may not be successful.

Problem-based learning (PBL) offers an alternative teaching approach to help motivate students to engage in authentic problem solving and to develop skills required for long life independent learning. Since its conception in North American medical schools a few decades ago, PBL has been adopted for the preparation of professionals in diverse fields such as engineering, law and business (Feather and Fry, 1999). PBL is however, mainly used within higher education institutions, and to date, it has yet to become a popular teaching approach at pre higher education level. Effective PBL starts with problems that can sustain students' interest as they attempt to reach a viable solution, and motivate them to probe for deeper understanding of the concepts being studied. Research has shown that engaging problems provide students with higher levels of comprehension and skill development (Albanese & Mitchell, 1993).

Since my initial literature review found that no models had been specifically developed for supporting software engineering education, my approach to developing a toolkit that supported problem-based learning in software engineering education was through four different case studies. The research method used for this study was case study and questionnaire.

3.2 Case studies using my pedagogical framework for underpinning small group collaborative learning

Consequently, in order to develop an appropriate pedagogical approach for underpinning small group collaborative learning, I developed a toolkit to enhance a previously developed Grid for scaffolding students learning within a problem-based learning environment (Oriogun and Georgiadou, 2000) to provide a structured representation for the kinds of activities undertaken by teaching agents in order to facilitate the development of new courses that included problem-based learning as part of their pedagogical model. Table 1 below shows the enhanced Problem-Based learning Grid **(Oriogun et al., 2002; Oriogun et al., 2003).**

My toolkit, the enhanced Problem-Based Learning Grid (or the ePBL Grid) was used as a framework for reflection in Studies 6 **(Oriogun, French and Haynes, 2002)** and 7 **(Oriogun et al., 2003)**. In Study 6, I investigated (with case studies from two multimedia colleagues at my University), the interaction of the course component of the Grid and the roles of students as both learners and multimedia developers; the interaction of the course component of the Grid with the roles and responsibilities of multimedia developer(s) and the course lecturers; and the final one is the course component of the Grid with some of the roles and responsibilities of the lecturer and the students in the development of online multimedia system.

In **(Oriogun et al., 2003)**, a subsection of the enhanced Problem-Based Learning Grid was used as a framework for reflection, when we adapted the Win-Win Spiral model for the development and documentation of software engineering projects at the London Metropolitan University. Six postgraduate masters software engineering students participated in Study 7 (three of whom were co-authors of the paper). It was argued in the paper that students could benefit from using the ePBL Grid to aid the documentation of software engineering artefacts when working in small teams collaboratively online, on campus and off campus within higher education institutions.

Course Component	Lecturer	Tutor	Multimedia Developer	Student
Lectures	• *Plan* • *Schedule* • *Liaise with tutors* • *Deliver*			• *Attend* • *Participate*
Online Resources	• *Liaise with developer* • *Design input* • *Provide content*	• *Facilitate* • *Support*	• *Project-manage* • *Design* • *Give pedagogical advice* • *Production of assets* • *Test prototype*	• *Interactivity*
Tutorials / Seminars / Workshops	• *Plan* • *Schedule* • *Allocate*	• *Organise* • *Facilitate* • *Monitor progress* • *Liaise with lecturer*		• *Contribute* • *Ask questions* • *Engage in problem solving* • *Report progress* • *Criticise (Peer)*
Computer Mediated Communication	• *Moderate and contribute* • *Set tasks*	• *Moderate and contribute* • *Set tasks*	• *Set up discussion groups* • *Set up chat rooms* • *Set up video conference*	• *Participate* • *Engage* • *Contribute*
Research	• *Suggest* • *Monitor* • *Evaluate*	• *Suggest* • *Help* • *Focus*		• *Plan activity* • *Research* • *Investigate* • *Document*
Individual Assignment	• *Specify* • *Monitor* • *Evaluate*	• *Direct* • *Monitor*		• *Research* • *Document* • *Present* • *Implement*
Teamwork	• *Specify task* • *Allocate groups* • *Liaise with tutors* • *Evaluate results*	• *Monitor* • *Assess Progress* • *Provide Feedback*		• *Research* • *Present results* • *Apply techniques* • *Implement software* • *Use tools* • *Participate* • *Deliver presentation* • *Peer review*
Formative Feedback	• *Liaise with developer* • *Provide content* • *Provide feedback*	• *Liaise with developer* • *Provide content* • *Provide feedback*	• *Design /develop online materials* • *Produce templates*	• *Self assessment* • *Self diagnosis* • *Peer assessment*
Summative Assessment	• *Plan* • *Write* • *Deliver*	• *Support revision*	• *Design /develop online materials* • *Technical support on security issues*	• *Prepare* • *Revise* • *Attend* • *Succeed*

Table 1. The enhanced Problem-Based Learning Grid –ePBL Grid **(Oriogun et al., 2002; Oriogun et al., 2003)**

The enhanced Problem-Based Learning Grid is a toolkit that provides a pedagogical framework and a documentation tool for underpinning small group collaborative learning online and offline.

4.0 Understanding the type of learning interaction that was taking place online

In order to understand the type of learning interaction in discussion groups that was taking place online with my software engineering students (my second research objective), I researched commonly measured variables when using content analysis. Commonly measured variables include participation, interaction, social elements, cognitive elements and meta-cognitive elements (Henri, 1992). Research on dialogue analysis has explored the relationship between online dialogue features (e.g., roles, strategies, form, and content) and learning (Pilkington 2001). Such an analysis can provide useful insights into the nature of the learning processes from the perspective of, for example, what a speaker's intention is in a transmitted message and what the receiver perceives has been communicated by the message.

However, problems can arise if one attempts to investigate specific categories or variables of the learning process - for example, participation, interaction, social elements, cognitive elements, and meta-cognitive elements (Henri 1992). In the case of coding protocols that include several categories, coders may not agree on interpretations. For this reason, some researchers (e.g., Potter and Levine-Donnerstein 1999) have argued that, although "the standard inter-rater reliability measure" Kappa (Cohen 1960) is powerful, it can be overly conservative.

Specifically, I examined interrater reliability measures of computer-mediated conferencing (Cohen, 1960; Holsti, 1969; Levine-Donnerstein, 1999) and suggested new coding categories relevant to problem-based learning for my own inter-rater reliability measure of online transcripts, called the Transcript Reliability Cleaning Percentage (TRCP) **Oriogun (2003).** I later developed the theoretical underpinning for my inter-rater reliability measure, with a colleague who provided a detailed literature review for the article **(Oriogun and Cook, 2003).** The variables that I measured using content analysis and grounded theory in Study 8 and 9 are *participation* and *interaction* of software engineering students online.

4.1 Case studies supporting the understanding of the type of learning interaction that had taken place online

In the academic year 2001-02, I tackled my third research objective in my Study 8. In Study 8, from a software engineering class of 95 students; a group consisting of 7 students was chosen to participate in the study. The group posted 141 messages over a period of 95 days. The unit of analysis for this study was message (Marttunen, 1977; Ahern, Peck and Laycock, 1992). The output from this study was my proposed inter-rater percentage agreement or inter-rater reliability measure of online transcripts using content analysis, called the Transcript Reliability Cleaning Percentage (TRCP) **Oriogun (2003).**

A grounded analysis of the transcripts produced a set of categories for describing the students' collaborations and learning interactions. This analysis led to the development of a structured approach to facilitating effective student interactions. Specifically, the output from this study led to the development of my theoretical framework for understanding small group collaborative learning. Table 2 shows coding decisions based on message ratings (for full details please see Section 9).

Coding Decision (Category)	Rating
No engagement with the group	0
Agreeing with others without reasons	1
Agreeing with others with reasons	2
Referring the group to relevant Web sites	3
Resolving conflicts within the group	4
Taking a lead role in discussion	5
Offering to deliver artefact(s)	6
Offering alternative solutions to group problems	7
Active engagement with the group	8

Table 2. Coding Decisions Based on Message Ratings (Oriogun, 2003a)

My Study 9 was from the same cohort of software students. A group consisting of 6 students was chosen to participate in the case study. The group posted 114 messages over a period of 64 days. The unit of content analysis for this study was message. The output from this study (addressed my research objective 3) was my methodological tool for analysing and understanding the interrater reliability measure of computer-mediated conferencing transcripts, called Transcript Reliability Cleaning Percentage (TRCP) **(Oriogun and Cook, 2003)**. Table 3 shows Coded Online Message Transcripts with Initial TRCP of 39%. Table 4 shows Coded Online Message Transcripts with Final TRCP of 87%. For more details, please read **(Oriogun and Cook, 2003)** from Appendix 1.

Student 1	Student 2	Student 3	Student 4	Student 5	Student 6
0, 5, 3,	0, 0, 8,	0, 6, 7,	4, 5, 8,	3, 7, 3,	3, 2, 3,
4, 3, 5,	2, 6, 3,	5, 7, 6,	5, 2, 8,	3, 3, 8,	3, 2
3, 3, 2,	3, 3, 5,	5, 3, 3,	4, 3, 3,	7, 5, 2,	
4, 2, 5,	5, 8, 3,	2, 2, 5,	3, 2, 6,	7, 2, 2,	
6, 3, 4,	5, 3, 4,	2, 5, 3,	0, 6, 5	5, 3, 5,	
3, 3, 3	5, 2, 2,	3, 6, 2,		3, 3, 2	
	3, 5, 5,	5, 2, 6,			
	5, 2, 2,	3, 5, 4,			
	2, 3, 3,	8, 5, 6,			
	8, 2	5, 3			
Total =18	Total = 29	Total = 29	Total = 15	Total = 18	Total = 5
Rating =3	Rating = 4	Rating = 4	Rating = 4	Rating = 4	Rating = 3

Table 3. Coded Online Message Transcripts with Initial TRCP of 39% **(Oriogun and Cook, 2003)**

Student 1	Student 2	Student 3	Student 4	Student 5	Student 6
0, 4, 3,	0, 0, 8,	0, 6, 8,	8, 8, 8,	3, 7, 3,	3, 6, 3,
8, 3, 5,	5, 7, 3,	5, 8, 6,	7, 5, 8,	5, 3, 8,	8, 2
3, 8, 2,	3, 3, 8,	7, 8, 3,	4, 3, 3,	7, 5, 2,	
7, 2, 5,	0,7, 3,	1, 2, 7,	3, 8, 6,	7, 2, 1,	
6, 3, 4,	7, 3, 4,	1, 8, 3,	0, 7, 8	2, 3, 5,	
3, 3, 3	8, 2, 2,	3, 6, 0,		3, 3, 2	
	8, 5, 8,	8, 0, 6,			
	8, 5, 2,	3, 5, 5,			
	5, 3, 3,	8, 5, 7,			
	8, 2	5, 3			
Total =18	Total = 29	Total = 29	Total = 15	Total = 18	Total = 5
Rating =4	Rating = 4	Rating = 5	Rating = 6	Rating = 4	Rating = 4

Table 4. Coded Online Message Transcripts with Final TRCP of 87%

(Oriogun and Cook, 2003)

Variables Investigated	Unit of Analysis (Message)	Final Rating Category*
None	No engagement with the group	LLE
Participation, Interaction	Agreeing with others without reasons	LLE
Participation, Interaction	Agreeing with others with reasons	LLE
Participation, Interaction	Referring the group to relevant Web sites	MLE
Participation, Interaction	Resolving conflicts within the group	MLE
Participation, Interaction	Taking a lead role in discussion	MLE
Participation, Interaction	Offering to deliver artefact(s)	HLE
Participation, Interaction	Offering alternative solutions to group problems	HLE
Participation, Interaction	Active engagement with the group	HLE

Table 5. Category of Final Student's Rating and Variables Investigated **(Oriogun and Cook, 2003)**

*MLE = Low Level Engagement
LLE = Medium Level Engagement
HLE = High Level Engagement

After understanding the type of learning that was taking place online through the development of my TRCP inter-rater reliability approach for measuring small group online collaborative 'participation' and 'interaction', I was able to further my research into online small group collaborative learning by adding 'cognitive elements' as another variable that merits investigation in the context of online small group CMC research. This cognitive dimension, led to my developing an appropriate method for analysing collaborative small group collaborative learning in software engineering education, which is called the SQUAD approach to online discourse.

I have also linked the work on TRCP to SQUAD through a journal article (Oriogun and Cook, 2003) and a conference paper (Oriogun, 2003a). In both publications, the theoretical basis of the SQUAD approach was mentioned as future extension to the TRCP research. It was further mentioned that the SQUAD approach will be a way of scaffolding online discourse, which we believe would make the coding of online messages much easier, avoiding the inter-rater reliability issues when analysing computer conference transcripts (Oriogun, Ravenscroft and Cook 2005:p230).

5.0 Developing an appropriate method for analysing collaborative small group learning in software engineering education online and off-line

My examination of interrater reliability measures of Computer-Mediated Conferencing lead to my alternative interrater reliability measure of online transcript called Transcript Reliability Cleaning Percentage (TRCP) **(Oriogun, 2003; Oriogun and Cook, 2003)**, and this helped me to make a smooth transition into developing an appropriate method for analysing collaborative small group learning in software engineering online, offline. Specifically, I was interested in addressing the following research question:

In what ways can we measure the quality of online learning levels of engagement with respect to *'participation', 'interaction' and 'cognition'?*

The difficulties of achieving acceptable levels of interrater reliability with CMC transcripts has sometimes lead to the development of semi-structured computer conferencing systems, in which participants choose the type of contribution that they are making from a limited set of alternatives. In order to identify an appropriate method for analysing collaborative learning in software engineering education, I conducted a literature review on existing semi-structured approaches to discourse analysis. The review revealed various ways of grouping online transcript messages (Austin, 1962; Searle, 1969; Baker, 1994; Duffy, Dueber, and Hawley, 1998; Cook, 2001; Pilkington, 2001; Rouke et al., 2001). After the review, in the 2002-03 academic year, I conducted my final Study 10 using 38 postgraduate software engineering students (the whole class participated in Study 9). For Study, I employed a combination of grounded theory and case studies to guide this research.

Strauss and Corbin (1990) suggest that grounded theory is especially useful for complex subjects or phenomena where little is yet known (as is the case in my study). This is because of the methodology's flexibility, which can cope with complex data and its continual cross-referencing, which allows for the grounding of theory in the data thus uncovering previously unknown issues.

The outcome of this research was the development of the new approach to CMC discourse. I suggested message-coding categories relevant to problem-based learning (Barrows, 1996; Bridges, 1992). This new approach encourages students to develop the skills of transferring knowledge into new domains, a skill that they can carry with them throughout their professional lives. This in turn empowers them with responsibilities of managing a largely self-directed learning process, and as a consequence, they are better equipped and informed to accept the responsibilities of mature professional life (Brine & Shannon, 1994). Foreman and Johnston (1999, p382) suggest that, "case studies can be based on real events in real organisations" (as it is in my Study 10). The new approach to measuring the quality of online discourse is called the SQUAD (**Oriogun, 2004**).

5.1 Development of a method for analysing collaborative small group learning in software engineering education online and offline

For study leading to the development of the SQUAD approach (**Oriogun, 2003**), it was very important to define what constitutes *cognition* with respect to the framework that I was developing for scaffolding online collaborative small group learning. According to (Ryder, 1994) knowledge is constructed by learners as they engage in dialogue. Furthermore, since the introduction of the "Zone of Proximity Development" (ZPD) continuum by Vygotsky (1962), it has been advocated by a number of authors that social interactions can act as scaffolding in the construction of knowledge. On the basis of this Vygotskian viewpoint, learning can be seen as a social phenomenon and experience. Table 6 shows the cognitive indicators' descriptors (please see Section 9 for more detail).

A number of theories on knowledge building emphasise the socially distributed nature of cognition. Distributed cognition is a process whereby individual cognition is extended to acquire something that an individual would be unable to achieve alone. Knowledge is constructed in associated networks of concepts and nodes. As learning occurs, new information is collected and coupled to existing knowledge networks. New information can then be easily retrieved to solve problems, and to apply in context.

Reasoning skills	Definitions
Elementary clarification	Observing or studying a problem, identifying its elements, and observing their linkages in order to come to a basic understanding.
In depth clarification	Analysing and understanding a problem to come to an understanding which sheds light on the values, beliefs, and assumptions which underlie the statement of the problem.
Inferencing	Induction and deduction, admitting or proposing an idea on the basis of its link with propositions already admitted as true.
Judgement	Making decisions, statements, appreciations, evaluations and criticisms. Sizing up.
Application of strategies	Proposing coordinated actions for the application of a solution, or following through on a choice or a decision.

Table 6: Cognitive Indicators Descriptors (adapted from Hara, Bonk & Angeli, 2000)

Students are expected to learn about the world based on their own research and study. Students determine their "knowns" and "unknowns". They seek knowledge to address their "unknowns". They engage in collaborative learning in their small groups to work on the problems (Wee, Kek & Sim, 2001, p159). Bruer (1993) argues that learning is quicker when students possess self-motivating skills generally referred to as meta-cognitive skills. Learning in PBL encourages meta-cognitive skills. In line with my usage of PBL, I adopted the adaptation of Henri's (1992) descriptors for "Reasoning Skills" as suggested by Hara, Bonk & Angeli, (2000) in support of cognitive indicators for my SQUAD approach **(Oriogun, 2003)**. Table 7 shows the descriptors for participation, interaction and cognition governing the new approach.

Message category	Description	Example	Cognitive indicators	Participation Indicators	Interaction indicators
S Suggestion	The process whereby the mere presentation of an idea to a receptive individual leads to the acceptance of the idea.	Students engage with other students within their coursework groups by offering advice, a viewpoint, or an alternative viewpoint to a current one.	-Elementary classification -In depth classification -Inferencing -Judgement -Application of strategies	Students engaging other students actively by taking a lead role in online discourse by posting meaningful and relevant messages to the group.	The message will be accessed and processed by other members of the group for the cycle of communication to be complete.
Q Question	A form of word address to a person in order to elicit information or evoke a response.	Students may seek clarification from the tutor or other students in order to make appropriate decisions relating to the group coursework.	-Elementary classification -In-depth classification	The message is posed in such a way that some or all the group members will engage in the ongoing discussion.	The message will be accessed and processed by other members of the group for the cycle of communication to be complete.
U Unclassified	Not in the list of categories of messages stipulated by the instigator of the task at hand.	This tends to happen at the start of the online postings. Students may be unsure of what the message is suppose to convey. In most cases, it falls within one of the four classified categories.	-Elementary classification	This type of message may or may not engage other students. In most cases, the message could be re-aligned to fall within the four classified categories by the coder of the final transcript at the end of semester.	This type of message may or may not engage other students. In most cases, the message could be re-aligned to fall within the four classified categories by the coder of the final transcript at the end of semester.
A Answer	Reply, either spoken or written, as to a question, request, letter or article.	Students are expected to respond to this type of message with a range of possible solutions / alternatives.	-Elementary classification -In depth classification -Inferencing -Judgement	Responding to a query or question will inevitably involve most, if not all the group members, especially if the response is not in line with other group members' opinions.	All group members are expected to deliver parts of the final product by working collaboratively.
D Delivery	The act of distribution of goods, mail etc.	Students are expected to produce a piece of software at the end of the semester. They all have to participate in delivering aspects of the artefacts making up the software.	-Elementary classification -In-depth classification -Inferencing -Judgement -Application of strategies	Each member of the group is expected to play an active role in delivering parts of the artefact making up the final software product. This is also expressed in the marking scheme for the module.	All group members are expected to deliver parts of the final product by working together collaboratively.

Table 7: The SQUAD approach to CMC discourse: Descriptors for participation, interaction and cognition **(Oriogun, 2003)**

5.2 The SQUAD software prototype

The SQUAD software prototype is the realisation of the SQUAD approach to online messaging. The architecture of the prototype is within my negotiated incremental architecture **(Oriogun, 2002; Oriogun and Ramsay, 2005).** The first version of the prototype was developed by one of my masters computing students as her dissertation (Small, 2003) for MSc Computing at London Metropolitan University. I later presented this first version at an international conference on information technology research in education **(Oriogun, 2004).** A second version of the SQUAD prototype has also been developed in-house within the department of Computing, Communication Technology and Mathematics at

London Metropolitan University, and we have reported the evaluation of the second prototype by Masters computing students through a special session of an international conference meeting on the integration of learning technologies with problem-based learning **(Oriogun and Ramsay, 2005)**. Figure 3 shows the SQUAD architecture.

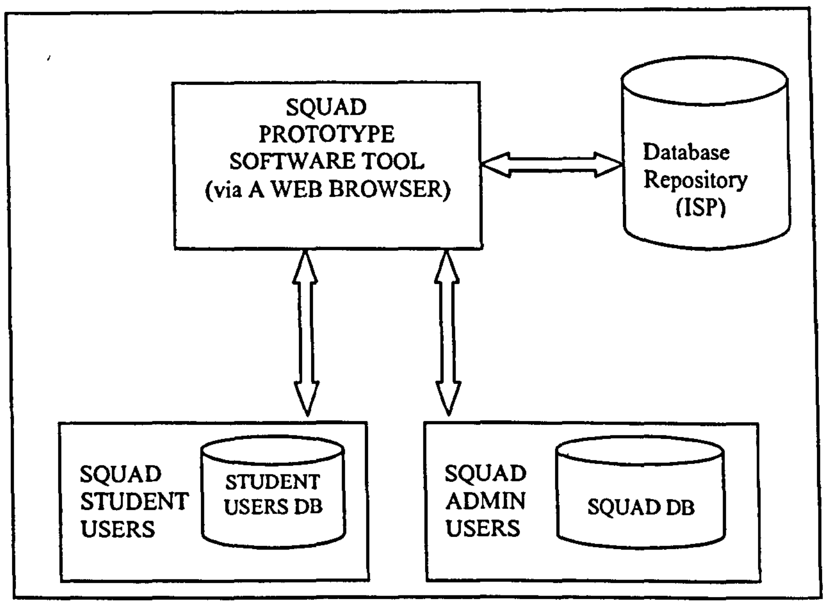


Figure 3. The SQUAD Architecture

The SQUAD v 2.0 software prototype supports the SQUAD approach to CMC discourse. It maintains groups of files and provides access to these files to users who will connect to the system from remote computers. The environment has a menu available so that students can easily navigate between the different options. There is a group registration facility with adequate protection such that only members of the group can view private information such as messages and files. Students are able to send private messages to members within their own group and others outside of their own group. Group members are able to read or download relevant files relating to the group coursework or upload new files. The application is able to calculate students 'online learning levels of engagement' (see **Oriogun, (2004)** for detail) as well as the statistics relating to the type / number of messages posted by each student within the two groups.

Only the Administrator has access to the all the SQUAD statistics, and all of the groups on the module. The Administrator is able to delete members within a group, or a whole group itself at the end of the module if required. The Administrator is the only person with the ability to delete posted messages. Figure 4 shows the login page for users, including the Administrator. The Administrator Interface is shown in Figure 5. Only the Administrator has access to SQUAD statistics (Oriogun and Ramsay, 2005).

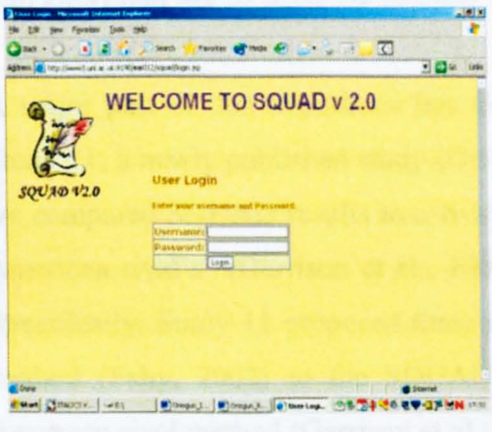


Figure 4: Main Login Page for SQUAD v 2.0

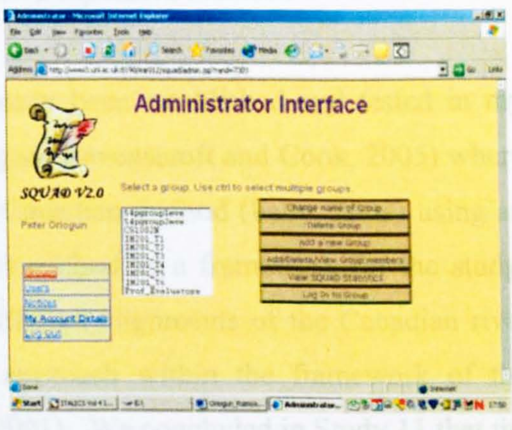


Figure5: Administrators Interface for SQUAD v 2.0

Figure 4 above shows all the functions that the Administrator is allowed. Figure 6 shows that a particular student belongs to two different groups within the system after login onto the system. Figure 7 shows the same user logged onto one of the groups she belonged to within the system (**Oriogun and Ramsay, 2005**).

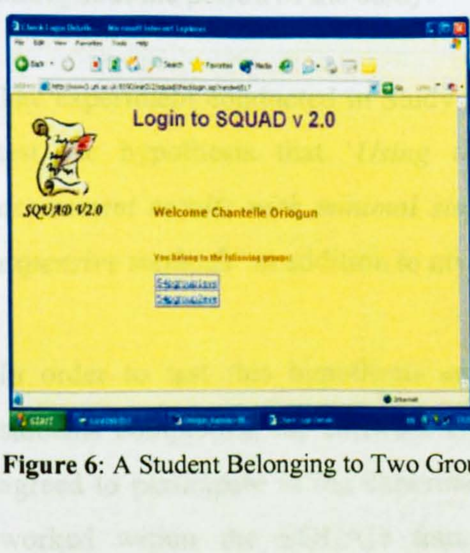


Figure 6: A Student Belonging to Two Groups

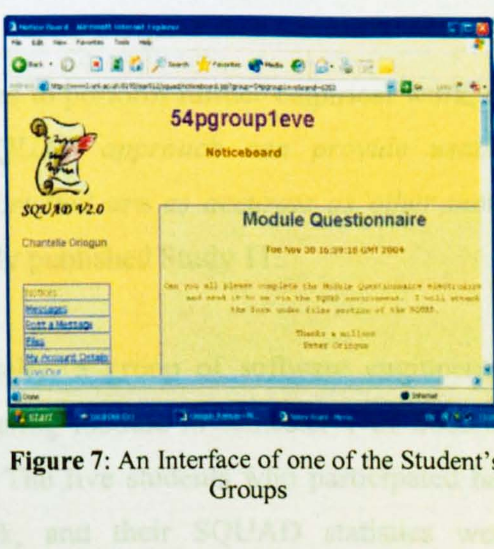


Figure 7: An Interface of one of the Student's Groups

6.0 Using the SQUAD Approach to Provide Usable Assessment Results with Minimal Staff Effort

In this section I conducted my Study 12, the experiment suggested by the two external examiners in order to make some quantifiable claims about the values of the SQUAD approach. The hypothesis tested here is as follows:

'*Using the SQUAD approach can provide usable assessment results with minimal staff effort that are as accurate as other more expensive methods*'

A major part of this hypothesis has already been established and tested in my Study 11; a newly published study (Oriogun, Ravenscroft and Cook, 2005) where we compared SQUAD results to a rival Canadian method (Fahy, 2002) using an American rival's (Garrison et al., 2001) method as a framework for the study. Specifically, Study 11 proposed three different alignments of the Canadian rival method (Fahy, 2002) to the SQUAD approach within the framework of the American rival method (Garrison et al., 2001). We concluded in Study 11 that the SQUAD approach proved to have shown much better results compared to the American rival (Garrison et el., 2001) method and the Canadian rival (Fahy, 2002) method (Oriogun, Ravenscroft and Cook, 2005: p197-214). Study 11 contained thirteen Masters software engineering students over the two semesters comprising the academic year 2004/05, and a grand total of 1039 messages were posted throughout the period of the study.

The experiment conducted in Study 12 was to perform further empirical work, to test the hypothesis that '*Using the SQUAD approach can provide usable assessment results with minimal staff effort that are as accurate as other more expensive methods*' in addition to my newly published Study 11.

In order to test this hypothesis empirically, a group of software engineering students completing the software engineering module in semester 1 of 2005/06 agreed to participate in the experiment. The five students who participated had worked within the SQUAD framework, and their SQUAD statistics were compiled through the SQUAD software prototype (Oriogun and Ramsay, 2005). These students contributed a grand total of 142-message postings over a period of 12 weeks comprising the semester. The number of message postings is more relevant in the context of this research rather than the number of students. The 142-message postings by the five students were considered to be an extension to Study 11, and consequently grounded the SQUAD approach using TRCP as a framework. The cognitive engagement of individual students within the group, and that of the group was measured using the mapping suggested in Study 11. Specifically Study 12 dealt with the following:

- The lecturer re-categorising undergraduate software engineering students' SQUAD messages using TRCP as a framework;
- Software engineering students re-categorising their individual SQUAD messages using TRCP as a framework;
- Grounding of SQUAD messages using TRCP as a framework;
- Software engineering students overall impression of their individual online message postings;
- To validate the consistency of the SQUAD approach and TRCP method results with respect to 'participation', 'interaction' and 'cognition'.
- Examining the group's cognitive engagement individually and as a group using the three alignments proposed in my newly published Study 11.

Table 8 below shows the number of SQUAD message postings by the group participating in this experiment throughout the semester. The statistics were generated using the SQUAD prototype, a proprietary in-house software tool supporting the SQUAD approach.

Student No	S	Q	U	A	D	TOTAL
Student 1	27	7	4	9	12	59
Student 2	14	6	4	6	8	38
Student 3	6	0	1	4	5	16
Student 4	3	3	2	2	2	12
Student 5	8	1	0	2	6	17
TOTAL	58	17	11	23	33	142

Table 8. Experimental Group's SQUAD Statistics
(Group and Individual SQUAD Contributions) Semester 1 - 2005/06

The associated online learning levels of engagement (Oriogun, 2003b) of each student is shown in Table 9 as calculated by the SQUAD prototype tool:

Student	High (%)	Nominal (%)	Low (%)
Student 1	66	15	18
Student 2	57	15	26
Student 3	68	25	6
Student 4	41	16	41
Student 5	82	11	5

Table 9. Experimental Group's SQUAD Online Learning Levels of Engagement

In the first semester of 2005/06, the five students were asked to be second coders (raters) of their own individual transcripts using data generated through the statistics compiled from the SQUAD software environment (see Table 8). It was expected that results obtained from such content analysis should be consistent with the students' online learning levels of engagement for each student as shown in Table 9.

The group posted 142 messages among its five students from 12th October 2005 until 11th January 2006. I extracted all the messages from this group in order to investigate the quality of each student's *participation* and *interaction* using *message* (Marttunen 1997; Ahern, Peck, and Laycock 1992) as a unit of analysis, where each message is objectively identified before producing a manageable set of cases that incorporates problem-based learning (Oriogun et al., 2002) activities before categorisation as documented in Table 2. It took a total of 5hours 45minutes to generate the initial TRCP values for all the transcripts as shown in Table 10. This exercise was conducted between 8th February 2006 and 15th February 2006 inclusive.

6.1 The TRCP Approach for this experiment

After carefully reading each of the 142 messages, I re-categorised and coded them (see Table 10 for the 'unclean' transcripts) using the criteria set out in Table 2. Each student was then rated according to the two variables being investigated, namely, *participation* and *integration* (see Table 5 for detail). Each student was asked to rate his or her own individual transcripts, generated when they used the SQUAD approach to negotiate software requirements online as a group in the first semester of 2005/06 (see Table 8).

Student 1 Initial TRCP = 56%	Student 2 Initial TRCP = 47%	Student 3 Initial TRCP = 13%	Student 4 Initial TRCP =8%	Student 5 Initial TRCP=47%
5,5,5,	8,5,5,	0,6,5,	0,8,0,	0,6,8,
7,7,8,	4,6,5,	0,6,6,	4,5,5,	5,6,5,
6,8,8,	7,8,6,	4,5,6,	5,4,0,	7,6,5,
5,8,7,	8,5,6,	6,6,6,	4,0,6.	5,6,5,
5,8,8,	5,5,8,	6,7,4,		6,6,5,
8,5,8,	8,8,5,	6.		6,6.
5,8,6,	5,6,5,			
5,6,5,	7,6,7,			
5,6,5,	6,6,6,			
6,5,5,	4,8,6,			
7,5,8,	7,7,6,			
5,5,7,	5,4,7,			
5,8,7,	8,8.			
7,7,7,				
7,7,5,				
7,6,6,				
5,5,6,				
2,8,6,				
6,8,8,				
2, 5.				
Total = 59	Total = 38	Total = 16	Total = 12	Total = 17
Rating = 6	Rating = 6	Rating = 5	Rating = 3	Rating = 6

Table 10. Experimental Group's Coded Online Message Transcripts with Initial TRCP Values (Unclean Transcripts)

The student coders (raters) also had access to the details in Table 2, as well as their individual transcripts from Table 8. Each student coder (rater) sought clarification from myself with respect to the rationale behind the categories of message ratings, and to fully understand the intention before generating their own set of ratings. It was not the duty of the student coders (raters) to convince the author to change his mind about the coding decisions. Once the student coders (raters) were satisfied that they understood the intentions behind each coding decision in Table 3, they rated the transcript independently, and eventually built their own compilation of ratings before the final TRCP was calculated (see Table 11).

6.2 Inter-rater Reliability Measure for this Experiment

Holsti (1969) provided the simplest and most common method of reporting inter-rater reliability—coefficient of reliability (C.R.)—as a percentage agreement statistic. The formula is

$$C.R. = 2m / n_1 - n_2$$

where: m= the number of coding decisions upon which the two coders agree

n_1 = number of coding decisions made by rater 1

n_2 = number of coding decisions made by rater 2

Cohen's Kappa (1960), on the other hand, is a statistic that assesses inter-judge agreement for nominally coded data. It can be applied at both the global level (i.e., for the coding system as a whole) and the local level (i.e., for individual categories). In either case, the formula is

$$Kappa = (F_0 - F_C) / (N - F_C)$$

where: N = the total number of judgements made by each coder

F_0 = the number of judgements on which the coders agree

F_C = the number of judgements for which agreement is expected by chance

A number of statisticians characterise the inter-judge agreement as inadequate, as it does not account for a chance agreement among raters (Capozzoli, McSweeney, and Sinha 1999). Therefore, with respect to Cohen's Kappa (1960), Capozzoli, McSweeney, and Sinha suggest that:

> ... values greater than 0.75 or so may be taken to represent excellent agreement beyond chance, values below 0.40 or so may be taken to represent poor agreement beyond chance, and values

between 0.40 and 0.75 may be taken to represent fair to good agreement beyond chance.(6)

6.3 Cleaning the Transcripts for this Experiment

In line with Capozzoli, McSweeney, and Sinha suggestion (Oriogun and Cook, 2003; pp227-228) I would further suggest that:

"...if the initial percentage agreement is >= 70%, the transcript is deemed to be "clean." In this case, the initial TRCP was the same as the final TRCP. Otherwise, a final TRCP should be calculated before the transcript can be considered to be "clean" and adequate given the subjectivity of such scoring criteria. The Kappa value (Cohen 1960) should be calculated from the clean transcript with a final TRCP."

Student 1 Final TRCP =100% Kappa = 1.0 $F_O = 59$ $F_C = 18$ N = 59	Student 2 Final TRCP =100% Kappa =1.0 $F_O = 38$ $F_C = 16$ N = 38	Student 3 Final TRCP =100% Kappa =1.0 $F_O = 16$ $F_C = 7$ N = 16	Student 4 Final TRCP=100% Kappa = 1.0 $F_O = 12$ $F_C = 11$ N = 12	Student 5 Final TRCP=100% Kappa = 1.0 $F_O = 17$ $F_C = 8$ N = 17
5,5,5,	5,5,5,	0,6,8,	0,5,8,	8,6,8,
4,7,8,	4,4,8,	2,6,4,	8,6,6,	5,6,7,
6,6,5,	5,8,8,	8,5,6,	8,5,8,	3,6,6,
5,8,5,	8,5,8,	6,7,6,	5,8,6,	5,6,6,
5,8,8,	5,5,8,	6,4,4,		8,6,8,
8,8,5,	8,8,5,	8.		6,6.
5,5,6,	5,6,8,			
5,6,4,	6,5,7,			
5,6,5,	6,6,6,			
6,5,5,	8,6,6,			
5,5,2,	7,5,6,			
5,5,5,	8,8,8,			
5,5,7,	8,8.			
7,7,7,				
2,5,5,				
6,6,6,				
5,5,6,				
1,8,6,				
6,8,6,				
1,5.				
Total = 59	Total = 38	Total = 16	Total = 12	Total = 17
Rating = 6	Rating = 6	Rating = 5	Rating = 6	Rating = 6

Table 11. Experimental Group's Coded Online Message Transcripts with Final TRCP Values (Clean Transcripts)

I invited the five students to the University on 17th February 2006 in order for each of them to rate their own transcripts before I calculated the initial TRCP values as shown in Table 10. Currently, Table 10 contains 'unclean' transcripts (Oriogun and Cook 2003, pp 226-227). I supplied the students with the coding decisions based on message ratings in Table 2, and told them that I have already used these categories to rate their SQUAD posted messages recently after they had completed their studies on the module, Software Engineering for Computer Science during the first semester of 2005/06.

I further explained the rationale behind each coding decision, and asked the students not to confuse themselves while rating their own online transcripts by thinking of the SQUAD approach to online discourse. When I was happy that all the students understood the intentions behind the coding schemata in Table 2, they were asked to individually rate their own transcripts. It took a total of 2 hours and 55 minutes to finalise the rating of all 142 online message transcripts after discussions with the two raters (students acted as second raters of their own transcripts as shown in Table 8, I acted as the first rater of each of the student transcripts) in order to generate the final TRCP value of 100, and a Kappa value of 1.0 for each student transcript on 17th February 2006 as shown in Table 11.

Once the transcripts have been 'cleaned' using the TRCP inter-rater reliability method, I used the phases of the *Practical Inquiry* model (triggers, exploration, integration and resolution) to realise the cognitive engagement of students in Group 2. Table 12 below shows the comparison of the phases of the *Practical Inquiry* model with the present Fahy (2005) *Practical Inquiry* / TAT results and Group 2 SQUAD results applying TAT alignments (Oriogun, Ravenscroft and Cook 2005, pp 205-210). See the concluding section for the analysis of Table 13.

6.4 The SQUAD TAT Alignments

Table 12 shows the proposed alignments of cognitive presence (Garrison, Anderson, and Archer 2000, 2001) in Oriogun's (2003b) SQUAD approach by adopting the TAT model (Fahy 2002) coding categories based on the TAT mapping (Oriogun, Ravenscroft, and Cook 2005: pp 201-205). Please note that the SQUAD alignments are such that, for each alignment, it is possible to have more than one of the categories of SQUAD within the four phases of the *Practical Inquiry* model.

Alignment	Triggers	Exploration	Integration	Resolution
#1	Q	U, S	A, S	S, D
#2	Q, A	U	S, D	S, D
#3	Q, A	U, S	S, D	S

Note: SQUAD = Suggestion, Question, Unclassified, Answer, Delivery.

Table 12. Proposed Alignments of Cognitive Presence (Garrison, Anderson, and Archer 2001, 2001) in Oriogun's SQUAD approach by adopting the Transcript Analysis Tool Model (Fahy 2002) Coding Categories (Oriogun, Ravenscroft, and Cook 2005:205)

Phases of the practical Inquiry Model	Practical Inquiry Model Results, Garrison, Anderson, and Archer (2001) Initial Pilot	Practical Inquiry Model Results, Fahy (2005) Present Study	TAT Results, Fahy (2005)	SQUAD Results Applying TAT Alignments SQUAD #1 Oriogun, Ravenscroft, and Cook (2005)	SQUAD Results Applying TAT Alignments SQUAD #2 Oriogun, Ravenscroft, and Cook (2005)	SQUAD Results Applying TAT Alignments SQUAD #3 Oriogun, Ravenscroft, and Cook (2005)
Triggers	12.5	9.4	6.4	11.8	28.2	28.2
Exploration	62.5	74.2	76.4	48.6	7.7	48.6
Integration	18.8	14.6	14.6	57.0	64.1	64.1
Resolution	6.3	1.8	2.5	64.1	64.1	40.1

Note. All table values are in percentages. TAT = Transcript Analysis Tool; SQUAD = Suggestion, Question, Unclassified, Answer, Delivery.

Table 13. Comparison of Phases of the Practical Inquiry Model With the Present Fahy (2005) Practical Inquiry/TAT Results and Experimental Group's SQUAD /TAT Alignments (Semester 1 - 2005/06)

In order to measure individual student cognitive engagement, Oriogun, Ravenscroft and Cook (2005) adopted the Practical Inquiry model as a framework within which they aligned SQUAD categories with the Transcript Analysis Tool (TAT) Fahy (2002) as shown in Table 12.

Phases of the practical Inquiry Model	SQUAD Results Applying TAT Alignments SQUAD #1 Oriogun, Ravenscroft, and Cook (2005)	SQUAD Results Applying TAT Alignments SQUAD #2 Oriogun, Ravenscroft, and Cook (2005)	SQUAD Results Applying TAT Alignments SQUAD #3 Oriogun, Ravenscroft, and Cook (2005)
Triggers	11.8	27.1	27.1
Exploration	52.5	6.8	52.5
Integration	61.0	66.1	66.1
Resolution	66.1	66.1	45.8

Note. All table values are in percentages

Table 14. Experimental Group's Student 1 SQUAD /TAT Alignments (Semester 1 –2005/06)

Therefore, in terms of an individual student's cognitive engagement based on their total number of postings over the semester, the average SQAD TAT alignment # 3 gives the best cognitive engagement for each student. The average SQUAD TAT alignment # 3 within the practical inquiry framework for Student 1 in Table 14 is 47.9% (average of column 4).

Phases of the practical Inquiry Model	SQUAD Results Applying TAT Alignments SQUAD #1 Oriogun, Ravenscroft, and Cook (2005)	SQUAD Results Applying TAT Alignments SQUAD #2 Oriogun, Ravenscroft, and Cook (2005)	SQUAD Results Applying TAT Alignments SQUAD #3 Oriogun, Ravenscroft, and Cook (2005)
Triggers	15.8	39.5	39.5
Exploration	47.4	10.5	47.4
Integration	52.6	57.9	57.9
Resolution	57.9	57.9	36.8

Note. All table values are in percentages

Table 15. Experimental Group's Student 2 SQUAD /TAT Alignments (Semester 1 –2005/06)

The average SQUAD TAT alignment # 3 within the practical inquiry framework for Student 2 in Table 15 is 45.4% (average of column 4).

Phases of the practical Inquiry Model	SQUAD Results Applying TAT Alignments SQUAD #1 Oriogun, Ravenscroft, and Cook (2005)	SQUAD Results Applying TAT Alignments SQUAD #2 Oriogun, Ravenscroft, and Cook (2005)	SQUAD Results Applying TAT Alignments SQUAD #3 Oriogun, Ravenscroft, and Cook (2005)
Triggers	0	25.0	25.0
Exploration	37.5	6.3	37.5
Integration	31.3	68.8	68.8
Resolution	68.8	68.8	37.5

Note. All table values are in percentages

Table 16. Experimental Group's Student 3 SQUAD /TAT Alignments (Semester 1 –2005/06)

The average SQUAD TAT alignment # 3 within the practical inquiry framework for Student 3 in Table 16 is 42.2% (average of column 4).

Phases of the practical Inquiry Model	SQUAD Results Applying TAT Alignments SQUAD #1 Oriogun, Ravenscroft, and Cook (2005)	SQUAD Results Applying TAT Alignments SQUAD #2 Oriogun, Ravenscroft, and Cook (2005)	SQUAD Results Applying TAT Alignments SQUAD #3 Oriogun, Ravenscroft, and Cook (2005)
Triggers	25.0	41.7	41.7
Exploration	41.7	16.7	41.7
Integration	41.7	41.7	41.7
Resolution	41.7	41.7	25.0

Note. All table values are in percentages

Table 17.Experimental Group's Student 4 SQUAD /TAT Alignments (Semester 1 – 2005/06)

The average SQUAD TAT alignment # 3 within the practical inquiry framework for Student 4 in Table 17 is 37.5% (average of column 4).

Phases of the practical Inquiry Model	SQUAD Results Applying TAT Alignments SQUAD #1 Oriogun, Ravenscroft, and Cook (2005)	SQUAD Results Applying TAT Alignments SQUAD #2 Oriogun, Ravenscroft, and Cook (2005)	SQUAD Results Applying TAT Alignments SQUAD #3 Oriogun, Ravenscroft, and Cook (2005)
Triggers	5.9	17.6	17.6
Exploration	47.1	0.0	47.1
Integration	58.8	82.4	82.4
Resolution	82.4	82.4	47.1

Note. All table values are in percentages

Table 18. Experimental Group's Student 5 SQUAD /TAT Alignments (Semester 1 – 2005/06)

The average SQUAD TAT alignment # 3 within the practical inquiry framework for Student 5 in Table 18 is 48.6% (average of column 4).

Phases of the practical Inquiry Model	SQUAD Results Applying TAT Alignments SQUAD #3 Oriogun, Ravenscroft, and Cook (2005) Student 1 Cognitive Engagement	SQUAD Results Applying TAT Alignments SQUAD #3 Oriogun, Ravenscroft, and Cook (2005) Student 2 Cognitive Engagement	SQUAD Results Applying TAT Alignments SQUAD #3 Oriogun, Ravenscroft, and Cook (2005) Student 3 Cognitive Engagement	SQUAD Results Applying TAT Alignments SQUAD #3 Oriogun, Ravenscroft, and Cook (2005) Student 4 Cognitive Engagement	SQUAD Results Applying TAT Alignments SQUAD #3 Oriogun, Ravenscroft, and Cook (2005) Student 5 Cognitive Engagement
Triggers	27.1	39.5	25.0	41.7	17.6
Exploration	52.5	47.7	37.5	41.7	47.1
Integration	66.1	57.9	68.8	41.7	82.4
Resolution	45.8	36.8	37.5	25.0	47.1

Table 18A: SQUAD Results Applying TAT Alignments SQUAD #3 Oriogun, Ravenscroft, and Cook (2005) Cognitive Engagement of Each Student using the Practical Inquiry Model as a Framework

Table 19 shows some of the actual messages sent by members from the Experimental Group under the S category of the SQUAD framework. See Appendix for these messages.

Message Number	Student Number	Final TRCP Rating
31	1	5
4	2	4
3	3	8
2	4	5
7	5	3

Table 19. Examples of Experimental Group's Students Message Postings and their Final Transcript Reliability Cleaning Percentage (TRCP) Values

6.5 Discussion of Results for this Experiment

The SQUAD statistics in Table 8 suggests that SQUAD Results Applying TAT Alignments SQUAD #3 Oriogun, Ravenscroft, and Cook (2005) using the *Practical Inquiry* model, as a framework is the best of the three alignments. The "High %" Online Level Learning Levels of Engagement realised within the SQUAD environment in Table 9 equates to the *Integration* within the Practical Inquiry mode. This means that we can now elaborate on other aspects of each student's cognitive engagement individually within their group with respect to the other three phases of the *Practical Inquiry* model, namely *Triggers*, *Exploration* and *Resolution.*

The results from this experiment suggest that the cognitive engagement of Student 1 (who posted 59 messages throughout the semester) and that of Student 5 (who posted a total of 17 messages throughout the semester) is literally the same (at 48% and 49% respectively), Student 2 (who posted 38 messages during the same period) closely followed with 45%, Student 3 (who posted 16 messages throughout the semester) average is 42% and finally Student 4 (who posted 12 messages) with an average of 36%.

These results suggest that the way the SQUAD approach measures cognitive engagement is not unfair, as someone who has made a good number of high quality interactions and also some general chat will not be marked or graded lower than someone who has made a low number of quality contributions.

6.6 Concluding Remarks for this Experiment

The fact that these five students had worked under the SQUAD framework, a semi-structured approach to online discourse before this exercise, during the formulation of the final TRCP values, meant the students became owners of their own transcripts, and were able to articulate the meaning and intentions behind each of their message postings. This is in fact why the final TRCP values for each of the students was indeed 100%.

It is argued in this section that the TRCP inter-rater reliability method has informed the SQUAD approach to online discourse through this empirical study. It is also claimed that it is possible to achieve a 100% inter-rater reliability measure at the message-level when grounding semi-structured CMC transcripts using an inter-rater reliability framework. It is further claimed that the SQUAD approach can provide usable assessment results with minimal staff effort that are as accurate as other more expensive methods such as its US and Canadian rivals.

6.7 Messages Sent by the Students in this Experiment

Student 1 Message 31
S-Student 3(Normalization + Process Model) – Student 1
Sun Nov 20 11:02:00 GMT 2005

Hi Student 3,

I saw your Normalization + Process model picture which are great. You need to change our ERD to reflect with your process model, which I find more complete.

However! I think you need to read just your normalisation. First we don't need customer details. One more, start with unnormalised set of data, then go to Normalisation 1 then 2 then you reach to level 3 which you have done..
Okie??
Before I forgot, please can you change the data in our zip database? What you need to change is in the Order Details table we have got Transaction date. Please can you change all 2003 into 2005 and keep the date and month.
Okie?
Cheers

Student 4 Message 2
S-Left members of the group – Student 4
Fri Oct 21 12:15:20 BST 2005

Hello all,

Some of us have already left the group and I don't know the name of them except Student X. Since we have to inform Peter how many people we need to replace asap, please post the name of the people who's left. I am sending emails to everyone in case those people who are already left won't see the SQUAD.

Student 5 Message 7
S-Lab – Student 5

Wed Nov 09 05:51:45 GMT 2005

Hey guys, I have a morning appointment 2day, which I unfortunately cannot reschedule, so I will not be able 2 make it to the lab 2day. If you guys can choose the features you decide to implement from the file I uploaded last week, I can finish work on the document. I will have the revised ERD up by tonite. Also, feel free to send me a task list of work, which we need to finish before the next meeting. Sorry once again.

Student 5

Student 3 Message 3
S-Important reading about winwin – Student 3

Fri Oct 14 23:38:22 BST 2005

I have uploaded the file for everybody; please make sure read it carefully. Should know about winwin negotiation before going to next steps. Other files will uploading soon.

Regards

Student 2 Message 4
S-TASK 2 – Student 2

Wed Oct 19 13:39:54 BST 2005

I've uploaded my work; sorry I didn't inform you all about not being in today. Not feeling well, happened overnight kinda thing, so apologies for not being there today at tutorial. I just quickly came on to send my work, its one part of it. The other 2 are a bit tricky.

Firstly, theres a bit about operational policies (policies on audit trails, copyright protection, etc), we haven't discussed that at all, so I have no idea what to put there.

Secondly, Operational stakeholders is very similar to effects of operations, since im basically writing the stakeholders involved, and how they interact etc, so I only had that under effects of operations.

And finally, redressal of current system shortfalls. We haven't talked about the proposed system, how it will be, what it will involve, so I don't know what to write for that. Only thing we know are the stakeholders. But we never went further than that to discuss how or what the proposed system would be like.

Ok, I think that's alot of reading.. but basically, at the end of the day, we can't just
keep going away like this and do tasks one after the other when the middle, or the end isn't clear. I don't know about the rest of you, but its like we're just trying to push along, without discussing how it's going to plan out at the end. Any comments would be appreciated.

7.0 Contribution

In my quest towards understanding and improving the process of small group collaborative learning in software engineering education, I developed an appropriate pedagogical approach for underpinning small group collaborative learning in the form of a toolkit. My enhanced problem-based learning grid (ePBL Grid) was developed specifically to facilitate the development of new courses that included problem-based learning as part of their pedagogical model. This minor contribution is a documentation toolkit for scaffolding learning by stipulating the roles and responsibilities of learning agents such as lecturers, students and multimedia developers in the first instance.

In order to understand the type of learning that was taking place with software engineering students in their small collaborative group online, it was important to analyse the communication between them. Specifically, this led me to carry out research into content analysis of online transcripts. Existing methods of analysing online transcripts were not adequate for my study; consequently, I developed my own method for measuring the online transcripts of software engineering students using grounded theory. My inter-rater reliability measure of online transcript, called Transcript Reliability Cleaning Percentage (TRCP) was first proposed in April 2003, and by December 2003; the theoretical underpinning for TRCP was developed.

This contribution extends previous work with respect to the inter-rater reliability measure of computer-mediated conferencing and suggested coding categories relevant to problem-based learning. Calculating inter-rater reliability agreement by using my Transcript Reliability Cleaning Percentage (TRCP) approach is simple for academics with a limited mathematical background and can provide insights into the nature of the learning process from the perspective of categorisation of online discourse.

My work on TRCP led to my major contribution to knowledge with respect to small group collaborative learning online, offline and distance education. Recent research in content analysis has shown the difficulties of achieving acceptable levels of inter-rater reliability with computer-mediated communication transcripts. This has lead to the development of semi-structured computer conferencing systems, in which participants choose the type of contribution that they are making from a limited set of alternatives.

The research demonstrates a thorough understanding of the type of learning interaction that was taking place with the software engineering students, and the problem that arises if we wished to investigate specific categories or variables of the learning process e.g. participation, interaction, social, cognitive and meta-cognitive elements. Specifically, if online interactions are to be transcribed and analysed using some theoretical framework, then the issue of coder interpretation at the time of coding a transcript becomes important.

In my major contribution to knowledge, I expanded on previous work with respect to semi-structured approaches to computer-mediated communication, suggesting coding categories relevant to problem-based learning. My SQUAD approach to online discourse offers definitions for quality with respect to participation, interaction and cognition, when using 'the message' as the unit of computer-mediated communication (CMC) transcript analysis, analysing what I have termed as 'online learning levels of engagement'. It is argued that the theoretical underpinning of the SQUAD approach is beneficial for the development of teamwork and cognitive reasoning when learning in small groups, and that it is a relatively straight forward exercise to apply this approach in a different mode of study or subject area.

It is claimed in the portfolio of work submitted that using the SQUAD approach can provide usable assessment results with minimal staff effort that are as accurate as other more expensive methods. In the portfolio of work submitted for the PhD, I have systematically investigated my research aim and problem in studies involving 241 different students over a period of 8 years. I contend in my submission that I have made a significant contribution to knowledge in my research in understanding and improving the process of small group collaborative learning in software engineering education within higher education, in order to prepare students for employment in software engineering by recommending (i) a toolkit for collaborative problem-based learning (ii) a methodological tool for analysing and understanding inter-rater reliability (iii) a framework for the development of teamwork and cognitive reasoning when learning in small groups.

8.0 Ongoing Work / Future Work

A second software prototype has been developed within the department of Computing, Communications Technology and Mathematics (CCTM), at London Metropolitan University to facilitate the SQUAD approach, and I have tested this prototype with over 130 software engineering students during the academic year 2004-05. It is my intention to seek funding to develop this current prototype to a marketable standard for use within the higher education sector. Future work will concentrate on online learning and problem-based learning, content analysis at the sentence level, software life cycle process models and pedagogical tools for scaffolding online learning.

9.0 References for covering documentation for PhD thesis by Prior Output

Ahern, T., K. Peck, and M. Laycock, M (1992). The effects of teacher discourse in computer-mediated discussion. *Journal of Educational Computing Research* 8 (3): 291–309.

Albanese, M., & Mitchell, S. (1993). Problem-Based Learning: A review of the literature on its outcomes and implementation issues. *Academic Medicine,* 68 (1), 52-81.

Austin, J. L. (1962). *How to do things with words.* New York: Oxford University Press.

Bagert, D. Hilburn, T., Hislop, G., Lutz, M., McCracken, M., and S. Mengel (1999). Guidelines for Software Engineering Education Version 1.0. Technical Report CMU/SEI-99-TR-032, Software Engineering Institute, Carnegie-Mellon University, October 1999.

Baker, M. J. (1994). A model for negotiation in teaching-learning dialogues. *Journal of Artificial Intelligence in Education,* 5(2), 199-254.

Barrows, H. (1996). Problem-based learning in medicine and beyond: A brief overview. In L. Wilkerson and W. Gijselaers (eds), *Bringing Problem-Based Learning to Higher Education: Theory and Practice.* New Directions for Teaching and Learning, 68, 3-11. San Francisco: Jossey-bass Publishers.

BCS (1989) British Computer Society and the Institute of Electrical Engineers. *A report on Undergraduate Curricula for Software Engineering.* London, England: Institute of electrical Engineers, June 1989.

Boehm, B W. (1996). Anchoring the Software Process, *IEEE Software,* vol. 13, no. 4, July 1996, pp73-82.

Boehm, B W. (1998). Using the WINWIN Spiral Model: A Case Study, *Computer*, vol. 31, no. 7, July 1998, pp 33-44

Boud, D., & Feletti, G. I.(1996). *The challenge of problem-based learning.* London: Kogan Page.

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

Bridges, E. M. (1992). *Problem-based learning for administrators.* ERIC Clearing House, University of Oregon.

Brine, J. & Shannon, S. (1994). Consolidating professional skills and developing the confidence of graduating students. In S. E. Chen, R. M. Cowdry, A. Kingsland & M. J. Ostwald (Eds), *Reflections on Problem Based Learning.* Sydney, Australia: Wild & Wooley Pty Ltd.

Bruer, H. S. (1993). *Schools for Thought: A Science of Learning in the Classroom.* Cambridge, MA: MIT Press.

Capozzoli, M., McSweeney, L. & Sinha, D. (1999). Beyond Kappa: A review of inter-ratter agreement measures. *The Canadian Journal of Statistics,* 27(1), 3-23.

Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educational and Psychological Measurements,* 20, 37-46.

Cook, J. 2001. Bridging the gap between empirical data on open-ended tutorial interactions and computational models. *International Journal of Artificial Intelligence in Education* 12: 85–99. Abstract available online at http://www.cogs.susx.ac.uk/ijaied/

Duffy, T., Dueber, B., & Hawley, C. (1998). Critical thinking in a distributed environment: A pedagogical base for the design of conferencing systems: In C. Bonk & K. King (Eds.). *Electronic contributors: learner-centered technologies for literacy, apprenticeship, and discourse (pp. 51-78).* New Jersey: Lawrence Erlbaum Associates, Publishers.

Foreman, J. & T. Johnston. 1999. Key aspects of teaching and learning in business and management studies. In H. Fry, S. Ketteridge, and S. Marshall (Eds), *A Handbook for Teaching and Learning in Higher Education,* 372-390. London: Kogan Page.

Fahy, P. J., G. Crawford, M. Ally, P. Cookson, V. Keller, and F. Prosser. 2000. The development and testing of a tool for analysis of computer mediated conferencing transcripts. *Alberta Journal of Education Research* 46 (1): Spring, 85–88.

Fahy, P. J. (2001). Critical thinking, cognitive presence, and computer conferencing in distance education. *The American Journal of Distance Education* 15 (1): 7–23.

Fahy, P. J. 2002. Assessing critical thinking processes in a computer conference. Centre for Distance Education, Athabasca University, Athabasca, Canada (unpublished paper). Available online at http://cde.athabascau.ca/softeval/reports/mag4.pdf

Fahy, P. J. (2005). Two methods for assessing critical thinking in computer-mediated communications (CMC) transcripts. *International Journal of Instructional Technology and Distance Learning* (March).
Available online at http://www.itdl.org/Journal/Mar_05/article02.htm

Feather, A. & H. Fry. (1999). Key aspects of teaching and learning in business and management studies. In H. Fry, S. Ketteridge, and S. Marshall (Eds), *A Handbook for Teaching and Learning in Higher Education,* 372-390. London: Kogan Page.

Ford, G A (1990). *1990 SEI Report on Undergraduate Software Engineering Education* (CMU/SEI-90-TR-003, ADA223881. Pittsburgh, Pa: Software Engineering Institute, Carnegie Mellon University, March 1990 [online] Available WWW <URL: http://www.sei.cmu.edu/publications/documents/90.reports/90.tr.003.html> (1990).

Garrison, D., Anderson, T. and Archer, W. (2001) 'Critical thinking, cognitive presence and computer conferencing in distance education'. *American Journal of Distance Education, 15*(1), 7-23. Available at http://www.atl.ualberta.ca/cmc/CTinTextEnvFinal.pdf (Accessed on 6th December 2004)

Gotterbarn, D, and Riser R (1994) *Real-World Software Engineering: A Spiral Approach to a Project-Oriented Course,* Proceedings, SEI Conference on Software Engineering Education, San Antonio, Texas, USA, 1994

Griffiths, S (1999). Key aspects of teaching and learning in business and management studies. In H. Fry, S. Ketteridge, and S. Marshall (Eds), *A Handbook for Teaching and Learning in Higher Education,* 95-107. London: Kogan Page.

Habra, N and Dubois E, (1994) *Putting into Practice Advanced Software Engineering Techniques through Students Projects,* Proceedings, SEI Conference on Software Engineering Education, San Antonio, Texas, USA, 1994

Hara, N., Bonk, C., & Angeli, C., (2000). Content analyses of on-line discussion in an applied educational Psychology course. *Instructional Science.* 28(2), 115-152.

Harrison, J V, (1997) *Enhancing Software Development Project Courses Via Industry Participation,* Presented at the 10th Conference on Software Engineering Education and Training, Virginia Beach, Virginia, USA, 1997

Henri, F. (1992). Computer conferencing and content analysis. In A. Kaye (Ed), *Collaborative learning through computer conferencing: The Najaden papers*, pp 117-136. London: Springer-Verlag.

Holsti, O. (1969). *Content analysis for social sciences and humanities.* Don Mills: Addison-Wesley Publishing Company. Working Knowledge Productive Learning at Work, International Conference, The Research into Adult and Vocational Learning Group, University of Technology at Sydney, New South Wales, Australia.

Hutton & Wiesenberg, (2000). Quality online participation: Learning in CMC classroom. *RCVET Working Knowledge Conference Papers.* Research Centre for Vocational Education and Training, University of Technology, Sydney, Australia, 10-13 Dec 2000. [viewed Mar 2003, verified 18 Sep 2003] http://www.rcvet.uts.edu.au/wkconference/working%20knowledge64.pdf

Howell-Richardson, C. & Mellar, H. (1969). A methodology for the analysis of patterns of participation within computer mediated courses. *Instructional science,* 24, 47-69.

Kerr, J and R Hunter. (1994). *Inside RAD*, McGraw-Hill, 1994.

Knolb, D. A. (1995). The process of experiential learning. In Thorpe et al. (Eds.), *Curriculum and Processes of Adult Learning.* Milton Keynes: The Open University.

Marttunen, M. 1997. Electronic mail as a pedagogical delivery system. *Research in Higher Education* 38 (3): 345–363.

Martin, J. (1991). *Rapid Application Development*, Prentice-Hall, 1991.

McDonald, J. (1998). Interpersonal group dynamics and development in computer conferencing: The rest of the story. In *Wisconsin Distance education Proceedings* Tri-Council Policy Statement: Ethical Conduct for Research Involving Humans. Available [online]: http://www.mrc.gc.ca/publications/publications.html

Neufeld, V and Barrows, H (1974) 'The McMaster Philosophy': an approach to medical education, *Journal of medical Education*, 49, pp 1040-50

Oriogun P K (1999). "A Survey of Boehm's Work on the Spiral Models and COCOMO II – Towards Software Development Process Quality Improvement", Software Quality Journal, 8, 53-62 (1999), Kluwer Academic Publishers, USA

Oriogun, P. K. & Georgiadou, E. (2000). Towards making learning agents think of their roles in the learning process through a problem-based learning grid. *Proceedings, 2nd Asia Pacific conference on PBL,* Singapore, 4-7 Dec 2000.

Oriogun P K (2001). "Towards Students' Negotiated Incremental Model -NIM: A Case Study on Students' In-Course-Assessment in a Software Engineering Module", Proceedings, UKAIS 2001 Conference, 18-20th April 2001, University of Portsmouth, UK, Zeus Press, ISBN 0-9540705-0-X

Oriogun P K, Gilchrist R (2002). "A Longitudinal Study on the Impact of Information Systems Analysis and Design Prerequisite on a Software Engineering Module", in the Proceedings (also available on CD-ROM), 7th Annual Conference of UKAIS 2002, Information Systems Research, Teaching and Practice, Leeds Metropolitan University, United Kingdom, 10th -12th April 2002, pp103-110, Published by Leeds Metropolitan University, ISBN 1 898883 149.

Oriogun P K (2002). "Towards Understanding Software Requirements Capture: Experiences of Professional Students using the NIA to Support the Win-Win Spiral Model", LTSN-ICS Electronic Journal (ITALICS), ISSN 1473-7507, Volume 1, Issue 2, August 2002.
[Online]: http://www.ics.ltsn.ac.uk/pub/italics/issue2/oriogun/007.html

Oriogun P K, French F, Haynes R (2002). " Using the enhanced Problem-Based Learning Grid: three multimedia case studies". In A. Williamson, C Gunn, A Young & T. Clear (eds), *Winds of*

Change in the Sea of Learning: Proceedings of the 19th Annual Conference of the Australian Society for Computers in Learning in Tertiary Education. Auckland, New Zealand: UNITEC Institute of Technology, 8th-11th December 2002, pp495-504, ISBN 0-473-09119-4.

Oriogun P. K (2003a). "Content Analysis of Online Inter-rater Reliability using the Transcript Reliability Cleaning Percentage ((TRCP): A Software Engineering Case Study", CD-ROM Proceedings, ICEIS 2003 (5th International Conference on Enterprise Information Systems), Angers, France, 23rd -26th April 2003, Volume 4, pp296-307, ISBN 972-98816-1-8.

Oriogun P K & Cook J (2003). "Transcript Reliability Cleaning Percentage (TRCP): An Alternative Inter-rater Reliability Measure of Message Transcripts in Online Learning", The American Journal of Distance Education, ISSN 0892-3647, 17(4), 221-234, December 2003

Oriogun P K, Khatri A, Choudhry Z, Borkhataria M (2003). "Using the Enhanced Problem-Based Learning Grid to Guide the Documentation of the Win-Win Spiral Model", vol 1, Proceedings, ASCILITE 2003 Conference (Australian Society for Computers in Learning in Tertiary Education), 7th-10th December 2003, Adelaide, Australia, pp 386-395, ISBN 0-9751702-0-1.

Oriogun P. K (2003b). "Towards understanding online learning levels of engagement using the SQUAD approach. Australian Journal of Educational Technology, 19(3), 371-388. http://www.ascilite.org.au/ajet/ajet19/ajet19.html

Oriogun P. K and Ramsay E (2005). "Introducing a dedicated prototype application tool for measuring students' online learning levels of engagement in a problem-based learning context", Proceedings, The IASTED International Conference on Education and Technology, ICET 2005, Calgary, Canada, July 4-6, 2005, pp329-334, CD-ROM ISBN: 0-88986-489-6, Book ISBN:0-88986-487-X.

Oriogun P K, Ravenscroft A and Cook J (2005). "Validating an Approach to Examining Cognitive Engagement within Online Groups", to appear in, The American Journal of Distance Education, ISSN 0892-3647, volume 19(4), 197-214, December 2005.

Piaget, J. "Judgement and reasoning in the chills", New York: Harcourt Brace, 1928.

Pilkington, R. (2001). Analysing educational dialogue interaction: Towards models that support learning (Introduction to the IJAIED Special Issue on Analysing Educational Dialogue). *International Journal of Artificial Intelligence in Education,* 12, 1-7.

Potter, W. & Levine-Donnerstein, D. (1999). Rethinking validity and reliability in content analysis, *Journal of Applied Communications Research*, 27, 258-284.

Pressman, R S. (2000). "Software Engineering -A Practitioner's Approach", 5th Edition, European Adaptation, pp 37-38, McGraw Hill 2000.

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

Rourke, L., T. Anderson, D. R. Garrison, and W. Archer (2001). Methodological issues in the content analysis of computer conference transcripts. *International Journal of Artificial Intelligence in Education* 12 (1): 8–22.

Rourke, L. & Anderson, T. (2004). 'Validity issues in quantitative computer conference transcript analysis', Educational Technology Research and Development 52(1) 5-18.

Royce, W. W. (1970). *Managing the development of large software systems: concepts and techniques,* Proc. WESCON, August 1970.

Ryder, M. (1994). Augmentation of the intellect: Network instruments, environments and strategies for learning. [viewed 28 Jan 2003, verified 18 Sep 2003] http://carbon.cudenver.edu/~mryder/augment.html

Searle, J. R. (1969). *Speech Acts: An essay in the philosophy of language.* Cambridge: Cambridge University Press.

Small N (2003). SQUADgroups. London Metropolitan University, Department of Computing, Communications Technology and Mathematics, North Campus, MSc Computing Dissertation, September 2003. http://www3.unl.ac.uk/~nas068[viewed on 24th March 2004]

Strauss, A. and Corbin, J. (1990). *Basics of Qualitative Research: Grounded Theory Procedures and Techniques.* Sage: London.

Trop, L. & Sage, S. (1998). *Problems as possibilities: problem-based learning for K-122* Virginia: ASDC

Vygotsky, L. S. (1962). *Thoughts and Language.* Cambridge, Mass: MIT.

Wee, K. N., Kek, Y.C., & Sim H. C. (2001). Crafting effective problems for problem based learning. In P. Little & P. Kandlbinder (Eds), *The Power of Problem Based Learning,* 157-168. ISBN 0 7259 11387. Australia: Australian Problem Based Learning Network.

Weiss, R. & Morrison, G. (1998). Evaluation of a graduate seminar conducted by listserv. [ERIC Document Reproduction Service, ED 423868]

Wilson B, Cockburn, J and Halligan, P (1987). "Development of a behavioral test of visuospatial neglect", Archive of physical medicine and rehabilitation, 1987 Feb; 68(2): 98-102 http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&list_uids=3813864&dopt=Citation [viewed 27the February 2005].

Zhu, E. (1996). Meaning negotiation, knowledge construction, and mentoring in a distance learning course. In Proceedings of Selected Research and Development Presentations at the 18th National Convention of the Association for Educational Communications and Technology, Indianapolis, IN. ERIC, ED 397849.

10.0 Remainder List of Research Output by Peter Kehinde Oriogun 1999 – 2005

10.1 Edited Books

Boyle T, Oriogun P. & Pakstas A (2004). 'Encouraging, Enabling and Equipping for Next Generation Developments', Proceedings of the 2nd International Conference Information Technology: Research and Education, London Metropolitan University, 28th June – 1st July 2004, IEEE Publication, ISBN 0-7803-8625-6 CD-ROM 0-7803-8626-4, IEEE Catalogue Number 04EX906, CD-ROM 04EX906C.

10.2 Book Chapters

Oriogun P K (2004). "SQUADgroups – A novel tool for capturing students' online learning levels of engagement with small groups", Proc ITRE 2004 (2nd International Conference Information Technology: Research and Education), T Boyle, P Oriogun and A Pakstas (eds), 28th June – 1st July 2004, London Metropolitan University, pp106-110, IEEE Publication, ISBN 0-7803-8625-6 CD-ROM 0-7803-8626-4, IEEE Catalogue Number 04EX906, CD-ROM 04EX906C.

Oriogun P K, Ferguson A W, Ludmann M, Maisuria N and Swhe Yu M (2001). "Using the Problem-Based Learning Grid in the Management of a Software Engineering Module: Towards an Effective Learning Intervention for Final Year Undergraduates", In, *The Power of Problem Based Learning,* Penny Little and Peter Kandlbinder (Eds.), 9th -12th December 2001, pp 99-110, Australian Problem Based Learning Network, C/o PROBLARC, The University of Newcastle, Callaghan NSW 2308, Australia, ISBN 0 7259 1138 7.

Cook J, Leathwood C, Oriogun P (2001). "Monitoring gender participation and promoting critical debate in an online conference", In, *Meeting at the Crossroads,* G. Kennedy, M Keppell & T Petrovic (eds.), Proceedings of the 18th Annual Conference of the Australian Society for Computers in Learning in Tertiary Education, pp.141-150, Melbourne: Biomedical Multimedia Unit, The University of Melbourne, Australia.
[Online] http://www.medfac.unimelb.edu.au/Ascilite2001/pdf/papers/cookj.pdf

10.3 Refereed Journal Publications

Cook J, Leathwood C, Oriogun P (2002). "Online Conferencing with Multimedia Students: Monitoring Gender Participation and Promoting Critical Debate", LTSN-ICS Electronic Journal (ITALICS), ISSN 1473-7507, Volume 1, Issue 2.
[Online]: http://www.ics.ltsn.ac.uk/pub/italics/issue2/cook/006.html

Oriogun P K (2000). "The Development of a Computing Module Incorporating Capability Curriculum", Educational Media International Journal, 37(4) 2000 ISSN 09523987.

10.4 Internationally Refereed Conference Publications

Oriogun P K (2006) "Towards understanding critical thinking process in a semi-structured approach to computer-mediated communication", To be presented at the Ed-Media 2006 Conference, Orlando, Florida, USA, 26th-30th June 2006.

Oriogun P K (2006) "Evaluating a proprietary software prototype for scaffolding students' online group learning", To appear in the Proceedings of the INSPIRE 2006 Conference, BCS Publication, Southampton Solent University, UK, 10th- 12th April 2006.

Oriogun P. K (2005). "Introducing a dedicated tool for facilitating a semi-structured approach to CMC messaging ", Workshop on e-Learning Online Communication, eLOC 2005, CD-ROM Proceedings of the third ACS/IEEE International Conference on Computer Systems and Applications (AICCSA-05) January 3-6, 2005 in Cairo, Egypt, IEEE Catalog Number 05EX949, ISBN 0-7803-8735-X, Library of Congress Number 2004110879.

Oriogun P. K (2003). "Transcript Reliability Cleaning Percentage (TRCP): An alternative Interrater Percentage Reliability of Online Discourse by Message Unit", Volume 4, Proceedings, ED-MEDIA 2003 Conference, Honolulu, Hawaii USA, June 23rd - 28th 2003, pp2608-2611, ISBN 380094-48-7.

Oriogun P K, Cook J (2003). "From Critical Debate to Content Analysis: Investigating the Impact of CMC on Campus-Based Students", 2nd International Conference on Multimedia and ICTs in Education (m-ICTE 2003), Badajoz, Spain, 3rd-6th December 2003, pp120-124, ISBN 84-96212-10-6.

Oriogun P K, Tilooh N S, Hansen J P, Assawaniwest P, Ranasinghe K N, Russell A and Anandasangaree R (2002). "Towards Software Negotiation Process Improvement: Experiences of Professional Students using the Win-Win Spiral Model", Proceedings, INSPIRE 2002 Conference, Limerick Institute of Technology, Ireland, 25th -27th March 2002, pp 91-106, BCS Publication, ISBN 1 902505 46 8.

Mikusauskas R and Oriogun P K (2002). "Module Evaluation, Redesign and Delivery using the Teaching and Learning Quality Chart", SQM 2002 Conference, Limerick Institute of Technology, Ireland, 25th -27th March 2002, pp 65-75, BCS Publication, ISBN 1 902505 46 8.

Cook J, Bradley C, Oriogun P, Henderson K and Baker A (2002). "ARTyFACTS: Designing an online learning environment for art objects", in the Proceedings, 11th International World Wide Web Conference, Honolulu, Hawaii, USA, 7th-11th May 2002. http://www2002.org/CDROM/alternate/691/index.html (CD-ROM: ISBN 1-880672-20-0).

Oriogun P. K (2002). "The effect of adopting iterative life cycle process models: case studies from undergraduate software engineering module", Proceedings, 13th International Conference on Information and Intelligent Systems", University of Zagreb, Varazdin, Croatia, 25th-27th September 2002, pp295-300, ISBN 953-6071-185.

Oriogun P K, Cook J (2001). Georgiadou E, "Professional Issues in a Software Engineering Module: Case Study Reports on Students' Coursework on Ethical Decision Making", Proceedings, 9th International Conference on Software Quality Management, SQM 2001, 18-20th April 2001, University of Loughborough, UK. British Computer Society Publication, ISBN 1-902505-40-9, pp 235-256.

Oriogun P K (2001). "The Effect of Adopting Iterative Life Cycle Process Models in Support of the Win-Win Spiral Model: Results of a Study on Students' Coursework in a Software Engineering Module", Proceedings, Software Quality Management, SQM 2001, 18-20th April 2001, University of Loughborough, UK. British Computer Society Publication, ISBN: 1-902505-40-9, pp325-341.

Oriogun P K (2001). Roberts W K, Zaouini M, Swann N J, "An Online Simulated Golf League Multimedia Environment: A Case Study from a Postgraduate Computing Module", Proceedings, Ed-Media 2001 Conference (Abstracts), 25-30th June 2001, Finland, ISBN 1-880094-42-8, ED-Media CD-ROM pp 1429-1431.

Cook J, Leathwood C, Oriogun P (2001). "Fostering Argumentative Skills and Monitoring Gender Contribution in Electronic Conferences", ILT 2001 Conference, 4-6th July 2001, University of York, UK.

Oriogun P K, Mikusauskas R (2001). "WinWin Decision Support Framework: A Case Study on Students' In-Course-Assessment in a Software Engineering Module", Proceedings, IEEE ICEIS 2001 Conference, 7-10th July 2001, Portugal, ICEIS Press, ISBN 972-98050-2-4, Vol 1, pp 370-376.

Oriogun P K and Henderson K J (2001). "Towards a Multimedia Learning Environment for the Construction Industry: Possible Design Considerations", Proceedings, AUBEA 2001 Conference, 11-13th July 2001, University of South Australia, Adelaide, ISBN: 0 86803 289 1, pp 93-98.

Henderson K J, Oriogun P K, Cook J and Baker A (2001). "Towards a Multimedia Learning Environment for Undergraduate Students Studying Architecture and Construction", Proceedings, (5th International Conference on Information Visualisation) IV2001 Conference, 25th-27th July 2001, University of London, England, IEEE Computer Society Order Number PR01195, Library of Congress Number 2001089877, ISBN 0-7695-1195-3, pp 621-623.

Cook J, Leathwood C and Oriogun P (2001). "Online Conferencing with Multimedia Students: Monitoring Gender Participation and Promoting Critical Debate", Proceedings, LTSN 2001 Conference, 27th-30th August 2001, University of North London, UK.

Oriogun P K (2000). "The Development of a Computing Module Incorporating Capability Curriculum", Proceedings (Web CD-ROM), Euro Education 2000, Aalborg, Denmark, 8th - 10th Feb '2000, ISBN 87-986760-1-6.

Oriogun P K (2000). "A Review of Theory-W Based Spiral Process Model and its Associated Tools for Supporting Conflict Identification and Negotiation in Software Development", Proceedings, SQM2000, 10th-12th April 2000, University of Greenwich, UK, BCS Publication, ISBN 1-902505-5, pp 53-64.

Oriogun P K and Gilchrist R (2000). "The Effect of Studying an Information Systems Module on Students' Results in a Software Engineering Module", Proceedings of the 5th UKAIS Conference, University of Wales Institute, Cardiff, 26th-28th April 2000, ISBN 007 709755 6, pp 394-401.

Oriogun P K and Cook J (2000)."A Capability Curriculum Framework: The Impact of Information Systems Prerequisite on a Software Engineering Module", Proceedings, INSPIRE V (Quality and Software Development: Teaching and Training Issues) Conference, 7-8th September 2000, British Computer Society Publication, ISBN 1-902505-36-0.

Oriogun P K and Georgiadou E (2000). "Towards Making Learning Agents Think of Their Roles in the Learning Process through a Problem-Based Learning Grid, 2nd Asia-Pacific Conference on Problem-Based Learning, Singapore, 4-7th December 2000. http://www.tp.edu.sg/pblconference/full/OriogunGeorgiadouPaper.pdf

Oriogun P K (1999). "A Survey of Boehm's Work on the Spiral Models and COCOMO II – Towards Software Development Process Quality Improvement", Proceedings, INSPIRE IV, Crete, 9th - 11th Sep '99, British Computer Society Publication, ISBN 1 90250516 6, pp123-133

Appendix 1: Hardcopies of Statements in relation to co-authors

The statements in this appendix relate to the prior output upon which this submission is based from each of the joint authors. These indicate and confirm the part that I played in each of the publications in terms of research and writing. The statements are from fellow professional research colleagues at London Metropolitan University, and my former undergraduate and postgraduate software engineering students who contributed to the case studies used in some of my publications.

These are cited in the following order:

Statement from Dr John Cook

Statement from Mr Ajeet Khatri

Statement from Professor Robert Gilchrist

Statement from Ms Fiona French

Second Statement from Dr John Cook

Statement from Dr Andrew Ravenscroft

Statement from Dr John Cook

Subject:
Re: Co-author Statements
Date:
Thu, 27 May 2004 15:03:47 +0100
From:
John Cook <john.cook@londonmet.ac.uk>
To:
Peter Oriogun <p.oriogun@londonmet.ac.uk>
References:
1

Dear Peter,
I confirm that the paper we jointly authored with Carole Leathwood, which resulted in the ITALICS article, involved you in the lead role in the interpretation of the statistical aspect of the paper and in commenting on my own reflections as part of the action research cycle.
The paper is cited below:

'Online conferencing with multimedia students: Monitoring gender participation and promoting critical debate', LTSN-ICS e-Journal (ITALICS), ISSN 1473-7507, vol 1 issue 2

I further confirm that with respect to the TRCP article, that Peter Oriogun was the driving force and major contributor to this and that my role was to review the literature on dialogue analysis and to read and comment on the paper with respect to his ongoing research into small group learning in the context of software engineering. The article is cited below:

'Transcript Reliability Cleaning Percentage (TRCP): An alternative interater Reliability Measure of Message transcripts in Online Learning', American Journal of Distance Education', 17(4) 221-234, ISSN 0892-3647, Online ISSN 1538-9286

I hope this fulfils the university requirements. Get back to me if further clarification is required.
Best Wishes,
John Cook

John Cook, Principal Research Fellow, Learning Technology Research Institute, London Metropolitan University http://homepages.unl.ac.uk/~cookj
Vice Chair, Association for Learning Technology http://www.alt.ac.uk
ALT-C 2004, 14-16 September 2004, Exeter http://www.alt.ac.uk/altc2004

Statement from Mr Ajeet Khatri

Subject:
Co-author Statements
Date:
Thu, 27 May 2004 20:31:58 +0100
From:
"AK Khatri" <ak44khatri@hotmail.com>
To:
"Peter Oriogun" <p.oriogun@londonmet.ac.uk>
References:
1

Dear Mr Oriogun,

I confirm on behalf of myself, Zaheeda Choudhry and Manish Borkhataria, that, you are the primary author with the driving force behind the paper we jointly authored as detailed below:

'Using the enhanced problem-based learning grid to guide the documentation of the Win-Win Spiral Model', Proceedings, ASCILITE 2003 Conference, pp386-395, ISBN CD-ROM 0-9751702-1-X WEB 0-9751702-2-8

Our role was a minor one, which was to provide a case study for the paper from our Software Engineering group coursework on the module IM54P (on the MSc Computing course at the London Metropolitan University) to guide the documentation the Win-Win Spiral model as part of your research into small group learning in the context of software engineering.

Ajeet Kumar Khatri

UNL ID: 02014185
MSc Computing

45 Abbotts Park Road
London E10 6HU

Phone: 07884 432 598

Statement from Professor Robert Gilchrist

Subject:
your request re the paper
Date:
Tue, 1 Jun 2004 10:41:23 +0100
From:
"R.Gilchrist" <r.gilchrist@unl.ac.uk>
To:
p.oriogun@unl.ac.uk
CC:
r.gilchrist@unl.ac.uk

Dear Peter,

I confirm that Peter Oriogun was the primary author and driving force behind the paper we jointly authored, as cited below:

'A longitudinal study on the impact of information systems analysis and design prerequisite on a software engineering module', Proceedings, UKAIS 2001 Conference, pp103-110, ISBN 1 898883 149

My role was to provide help with the statistical aspects of the paper, which concerned Peter's research into small group learning in the context of software engineering.

Bob Gilchrist

Professor R Gilchrist
Director, STORM,
The Statistics, Operational Research and Mathematics Research Centre of
London Metropolitan University
166-220 Holloway Road
London N7 8DB

+++++===========IMPORTANT UPDATE: PLEASE NOTE=========+++++++

The University of North London merged on 1st August 2002 with London Guildhall University to form
London Metropolitan University.

All my work will continue exactly as before, but I would be grateful if you would note the new designation, and the new e-mail address that I will be using - r.gilchrist@londonmet.ac.uk

The old e-mail will continue to work for a limited period, but please update your address book as soon as convenient

Thank you.

Statement from Ms Fiona French

Subject: PhD
From: "Fiona French" <fiona@msfiat.com>
Date: Wed, June 30, 2004 8:33 am
To: p.oriogun@londonmet.ac.uk (more)
Cc: fiona@msfiat.com
Priority: Normal
Options: View Full Header | View Printable Version

To whom it may concern:

From May to August 2002, I worked with Richard Haynes and Peter Oriogun to develop Peter's problem-based learning grid so that it encompassed multimedia aspects. Peter was the driving force and major contributor to the paper.

The experience was both enjoyable and a useful piece of collaborative research.

The paper dealt with 3 case studies. One of these was my experience of working with teams of students who were given the task of designing and developing a computer game - an open-ended problem and one that required them to work together and use all their skills. I wrote about my case study for the paper and also contributed to the overall paper by proof reading and offering minor edits.

Richard Haynes supplied a second case study for the enhanced grid paper: "Multimedia Development of the IncoChallenge". Peter supplied the third case study from a postgraduate software engineering coursework.

We were able to use the three case studies assembled to inform the enhanced grid. The case studies meant that the enhanced grid could be tested against existing multimedia examples. The purpose of the paper was to contribute to Peter Oriogun's research into small group learning in the context of software engineering.

--

Fiona French
Senior Lecturer in Multimedia
London Metropolitan University
e: fiona@msfiat.com / f.french@londonmet.ac.uk

Message sent using UebiMiau 2.7.2

Second Statement from Dr John Cook

-------- Original Message --------
Subject: Re: Co-authorship statement
Date: Mon, 20 Feb 2006 15:36:27 +0000
From: John Cook <john.cook@londonmet.ac.uk>
To: Peter Oriogun <p.oriogun@londonmet.ac.uk>
References: <43F9E0FE.7080109@londonmet.ac.uk>

Dear Peter,
I confirm that you were the major contributor to this article.
Regards John

Peter Oriogun wrote:

Dear Andrew and John,

I have submitted eleven of my publications for a PhD by prior output to London Metropolitan University and have put forward a portfolio of publications that I have authored and jointly authored. One of these I jointly wrote with both of you. The article was requested to be one of my submissions by my two external examiners on 6th February 2006, after my viva voce examination.

The article is as follow:

"Validating an Approach to Examining Cognitive Engagement Within Online Groups", The American Journal of Distance Education, 19(4), 97-214, Lawrence Erlbaum Associates, Inc

The university needs a statement from both of you say that I was the major contributor to this article. If you are able to provide this it would be much appreciated, and it should be okay to provide this, by simply responding to this email and this will give the University appropriate evidence. You may wish to elaborate further in your reply.

Best wishes and I look forward to hearing from you.
--

Dr John Cook
Reusable Learning Objects CETL Centre Manager
Office: 020 7133 4341
Contact details:
http://homepages.north.londonmet.ac.uk/~cookj/#contact

--
Peter Oriogun
****
Course Leader MSc Computing
London Metropolitan University
North Campus
2-16 Eden Grove
London N7 8EA
Direct Line (Work): 0207 133 7065 Fax (Work): 0207 133 7053
Email: p.oriogun@londonmet.ac.uk
Home: oriogunp@onetel.com
http://www.londonmet.ac.uk/depts/cctm/staff-pages/peter_oriogun.cfm
****
Honorary Visiting Lecturer in Software Engineering
Cranfield University
Bedford MK43 0AL

Statement from Dr Andrew Ravenscroft

-------- Original Message --------
Subject: Re: Co-authorship statement
Date: Mon, 20 Feb 2006 17:25:55 -0000 (GMT)
From: Andrew Ravenscroft <a.ravenscroft@londonmet.ac.uk>
Reply-To: a.ravenscroft@londonmet.ac.uk
To: Peter Oriogun <p.oriogun@londonmet.ac.uk>
CC: Andrew Ravenscroft <a.ravenscroft@londonmet.ac.uk>, John Cook <john.cook@londonmet.ac.uk>
References: <43F9E0FE.7080109@londonmet.ac.uk>

Dear Peter and John,

I am happy to confirm that you, Peter Oriogun, was the major contributor to this article.

Let me know if you need anything else.

Regards,

Andrew

Dear Andrew and John,

I have submitted eleven of my publications for a PhD by prior output to London Metropolitan University and have put forward a portfolio of publications that I have authored and jointly authored. One of these I jointly wrote with both of you. The article was requested to be one of my submissions by my two external examiners on 6th February 2006, after my viva voce examination.

The article is as follow:

"Validating an Approach to Examining Cognitive Engagement Within Online Groups", The American Journal of Distance Education, 19(4), 97-214, Lawrence Erlbaum Associates, Inc

The university needs a statement from both of you say that I was the major contributor to this article. If you are able to provide this it would be much appreciated, and it should be okay to provide this, by simply responding to this email and this will give the University appropriate evidence. You may wish to elaborate further in your reply.

Best wishes and I look forward to hearing from you.

Peter Oriogun
Course Leader MSc Computing, London Metropolitan University
North Campus, 2-16 Eden Grove, London N7 8EA
Direct Line (Work): 0207 133 7065 Fax (Work): 0207 133 7053
Email: p.oriogun@londonmet.ac.uk Home: oriogunp@onetel.com
http://www.londonmet.ac.uk/depts/cctm/staff-pages/peter_oriogun.cfm
Honorary Visiting Lecturer in Software Engineering
Cranfield University
Bedford MK43 0AL

Dr. Andrew Ravenscroft (C.Psychol, PhD, MSc, BSc)
Deputy Director, Principal Research Fellow
Learning Technology Research Institute (LTRI), London Metropolitan University
Room 204, Shoreditch Building, 35 Kingsland Road, London E2 8AA
a.ravenscroft@londonmet.ac.uk
http://homepages.unl.ac.uk/~ravensca/
Tel: +44 (0)20 7749 3753
Fax: +44 (0)20 7749 3781

Appendix 2: Hardcopies of Prior Output upon which PhD thesis is based

The prior output upon which PhD is based are cited in the following order in Appendix 1:

Oriogun P K (1999). "A Survey of Boehm's Work on the Spiral Models and COCOMO II – Towards Software Development Process Quality Improvement", Software Quality Journal, 8, 53-62 (1999), Kluwer Academic Publishers, USA

Oriogun P K (2001). "Towards Students' Negotiated Incremental Model -NIM: A Case Study on Students' In-Course-Assessment in a Software Engineering Module", Proceedings, UKAIS 2001 Conference, 18-20th April 2001, University of Portsmouth, UK, Zeus Press, ISBN 0-9540705-0-X

Oriogun P K & Gilchrist R (2002). "A Longitudinal Study on the Impact of Information Systems Analysis and Design Prerequisite on a Software Engineering Module", in the Proceedings (also available on CD-ROM), 7th Annual Conference of UKAIS 2002, Information Systems Research, Teaching and Practice, Leeds Metropolitan University, United Kingdom, 10th -12th April 2002, pp103-110, Published by Leeds Metropolitan University, ISBN 1 898883 149.

Oriogun P K (2002). "Towards Understanding Software Requirements Capture: Experiences of Professional Students using the NIA to Support the Win-Win Spiral Model", LTSN-ICS Electronic Journal (ITALICS), ISSN 1473-7507, Volume 1, Issue 2, August 2002.
[Online]: http://www.ics.ltsn.ac.uk/pub/italics/issue2/oriogun/007.html

Oriogun P K, French F and Haynes R (2002). " Using the enhanced Problem-Based Learning Grid: three multimedia case studies". In A. Williamson, C Gunn, A Young & T. Clear (eds), *Winds of Change in the Sea of Learning*: Proceedings of the 19th Annual Conference of the Australian Society for Computers in Learning in Tertiary Education. Auckland, New Zealand: UNITEC Institute of Technology, 8th-11th December 2002, pp495-504, ISBN 0-473-09119-4.

Oriogun P. K (2003). "Content Analysis of Online Interrater Reliability using the Transcript Reliability Cleaning Percentage ((TRCP): A Software Engineering Case Study", CD-ROM Proceedings, ICEIS 2003 (5th International Conference on Enterprise Information Systems), Angers, France, 23rd -26th April 2003, Volume 4, pp296-307, ISBN 972-98816-1-8.

Oriogun P K and Cook J (2003). "Transcript Reliability Cleaning Percentage (TRCP): An Alternative Inter-rater Reliability Measure of Message Transcripts in Online Learning", The American Journal of Distance Education, ISSN 0892-3647, 17(4), 221-234, December 2003

Oriogun P K, Khatri A, Choudhry Z and Borkhataria M (2003). "Using the Enhanced Problem-Based Learning Grid to Guide the Documentation of the Win-Win Spiral Model", vol 1, Proceedings, ASCILITE 2003 Conference (Australian Society for Computers in Learning in Tertiary Education), 7th-10th December 2003, Adelaide, Australia, pp 386-395, ISBN 0-9751702-0-1.

Oriogun P. K (2004). "Towards understanding online learning levels of engagement using the SQUAD approach. Australian Journal of Educational Technology, 19(3), 371-388. http://www.ascilite.org.au/ajet/ajet19/ajet19.html [available from 23rd January 2004]

Oriogun P. K and Ramsay E (2005). "Introducing a dedicated prototype application tool for measuring students' online learning levels of engagement in a problem-based learning context", Proceedings, The IASTED International Conference on Education and Technology, ICET 2005, Calgary, Canada, July 4-6, 2005, pp329-334, CD-ROM ISBN: 0-88986-489-6, Book ISBN:0-88986-487-X.

Oriogun P K, Ravenscroft A and Cook J (2005). "Validating an Approach to Examining Cognitive Engagement within Online Groups", to appear in, The American Journal of Distance Education, ISSN 0892-3647, 19(4), 197-214, December 2005.

Software Quality Journal, 8, 53–62, 1999

A Survey of Boehm's Work on the Spiral Models and COCOMO II—Towards Software Development Process Quality Improvement

PETER K. ORIOGUN p.oriogun@unl.ac.uk
School of Informatics and Multimedia Technology, University of North London, London N7 8EA

Abstract. Successful engineering and evaluation of complex software depends on successfully completing all the stages of the Software Development Life Cycle. There have been many models which illustrate the stages Software Engineers have to go through to produce software. This paper investigates one of these—The Spiral Model (Sommerville 1997) with particular reference to recent enhancements to it, examines the interaction of COCOMO II (1998) within the WinWin framework, and reports on a case study using the WinWin Spiral Model (Boehm et al. 1998) to develop software.

Keywords: COCOMO II, WinWin spiral model, theory-W

1. Introduction

The language and technology of the Information Technology world is still developing. There exists no real maturity of understanding as yet about limits or capabilities. This will only come when people refer to the IT revolution in the past tense. In the 1960's software development was initially regarded, as if it were a science art form, with applications being fairly limited, and methodologies not being closely monitored. As hardware has developed thus allowing for more sophisticated software development costs have risen sharply.

The Software Engineer attempts to bring a piece of software to the customer that meets explicitly the customer's requirements, ensuring that this process is cost effective by maximising efficiency. To this end, as the discipline of software engineering has developed, a number of "software engineering" models and constructs have evolved to define and refine the process. Models such as the Waterfall (Royce 1970), the Spiral Model (Sommerville 1997) and evolutionary variants (Boehm et al. 1999) should go a long way in modelling software development by actively engaging its stakeholders.

2. The spiral model

Any project presents certain levels of risk; however it is the ability to manage these risks that ultimately dictates the success of a given task. Early attempts at the software development, such as the waterfall model, did not fully embrace the impact these risks might have on the development process. Major software projects failed because project risks were neglected and nobody was prepared when something unforeseen happened. The Spiral Model is an evolutionary software process model (Boehm 1998) that combines the iterative aspect of prototyping with the controlled and systematic nature of the linear sequential model. The *spiral model* uses risk analysis (Pressman 1998) as one of its elements (this also helps in being more compatible with the other models). The ambitious aim of the new model was to incorporate the strengths and avoid the difficulties of the other models by shifting the management emphasis to risk evaluation and resolution.

The spiral model is not as sophisticated and elaborate as other software lifecycle models. It requires further planning and development in such areas as contracting; specifications; milestones, reviews, scheduling, status monitoring and risk identification (Boehm et al. 1994).

3. The WinWin spiral model

The WinWin framework (Boehm et al. 1994) extends the original spiral model by adding Theory-W activities to the front of each cycle.

3.1. Theory-W

Theory-W (Boehm et al. 1994) is designed for general application and is not confined to software development. The stakeholders are defined as:

- Users, customers, developers, maintainers, interfacers, testers, re-users, general public

Essentially the theory argues that a project will only succeed if the critical stakeholders (users, customers, developers and maintainers) are all "winners;" thus the term WinWin. If the requirements of any one of these stakeholders are omitted, then a Win-Lose situation pertains—for example, where the customer's requirements are met but not those of the users.

Such a Win-Lose scenario is, in reality, a Lose-Lose situation because a project can only succeed if all major parties achieve their essential objectives. A typical example could be a situation whereby a piece of software meets the customer's cost requirements but doesn't actually work according to the user's requirements. The "winner" also loses since s/he is going to have to either put up with a less than optimal performance or be faced with further expense of developing the software.

3.2. Next Generation Process Model (NGPM)

The Next Generation Process Model (Boehm 1995) is software that sorts through the different win conditions imposed by the multiple stakeholders, then gives an

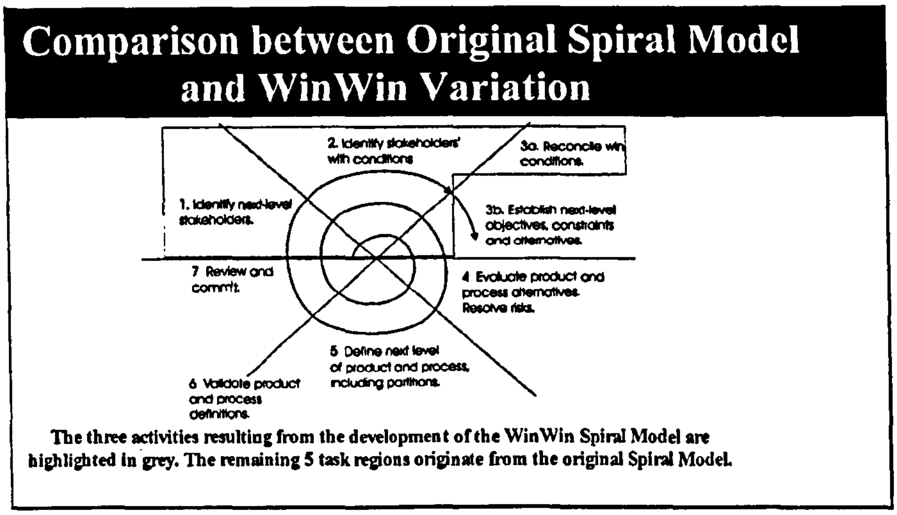


Figure 1. The WinWin Spiral Model (Boehm 1998).

approach that is suitable and beneficial to all. It uses the Theory-W approach to unite on a system's next level objectives, constraints and alternatives (1998). The NGPM also categorically addresses the need for simultaneous analysis, risk solution, definition and elaboration of both the software product and the software process.

Software organisations needed to have a common frame of reference as a basis for organising management procedures, defining the cost of the project and estimating a schedule for completion, and for communicating with other organisations. Their need for such a common frame of reference was such that they often have stuck with flawed models (e.g. the waterfall model) just for the sake of having a common framework. Three process milestones have been identified (Boehm et al. 1994), which anchor down the software process, allowing to maintain the necessary flexibility but adding discipline and focus. The three anchor points are the Life-Cycle Objectives (LCO), the Life-Cycle Architecture (LCA) and the Initial Operational Capability (IOC).

4. WinWin decision model

The WinWin process is modelled using four main objectives (Boehm 1998):

Win Condition, Issue, Option and Agreement

Figure 2 shows a typical structure of the decision rationale in terms of the above elements and the link types showing the relation between them (Bose 1995).

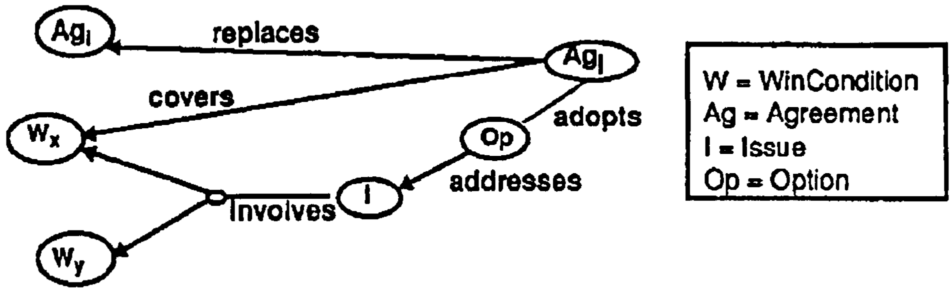


Figure 2. WinWin decision objects and relations between them (Bose 1995).

4.1. *WinWin support framework*

Figure 3 shows a diagram of the support framework. It supports the WinWin concept of collaboration (Bose 1995).

4.2. *WinWin Tools*

WinWin is a Unix workstation based computer program that aids the capture, negotiation and co-ordination of requirements for a large system. It assumes that a group of people called stakeholders have signed on with the express purpose of discussing and refining the requirements of their proposed system (Horowitz 1999).

WinWin-0, the initial version of the support system, was implemented on top of Perception CAGE/PM®. A "bootstrap experiment" was performed at University of Southern California Centre for Software Engineering (USC-CSE) with WinWin-0 by using it to model the next version of University of Southern California (USC) own WinWin product.

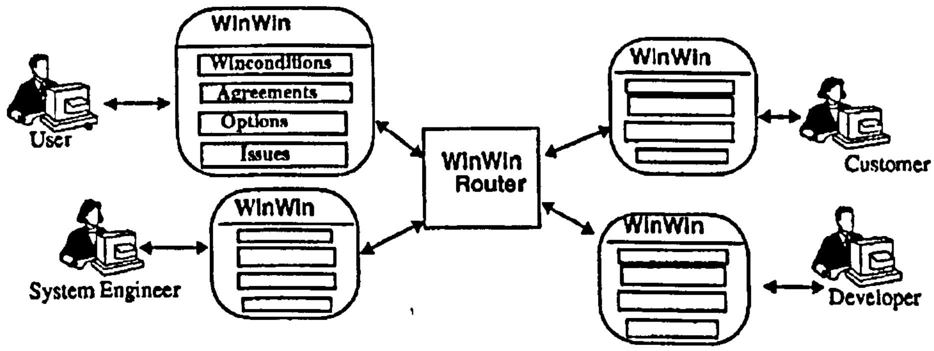


Figure 3. A systems diagram of the WinWin architecture (Bose 1995).

5. COCOMO 81

The COCOMO 81 model allows one to estimate and predict software development cost, effort and schedule via point estimates when planning a new software development activity, according to software development procedures which were commonly used between the 1970's and 1980's (1999). It consists of three forms, each form offering greater detail and accuracy the further along one is in the project planning and design process.

The instrumented tool provides cost, effort and schedule point estimates. It also allows the planner to freely perform "what if" scenario investigation, by quickly highlighting the effect adjusting requirements, resources and staffing might have on estimated costs and schedules. COCOMO 81 has 63 data points (Sommerville and Sawyer 1997).

5.1. COCOMO II

COCOMO (1999) is a model, which allows an individual to calculate the cost, effort and schedule when planning a new software development exercise. It consists of three sub-models, each one offering increased devotion the further along one is in the project planning and design process. Listed in increased devotion, these sub-models are:

- Applications Composition sub-model
- Early Design sub-model
- Post–architecture sub-model

Only the last (and also the most detailed) sub-model, Post-architecture, has been implemented in a regulated software tools. The Post-architecture sub-model is also referred to as COCOMOII. USC-COCOMOII user manual (1999) documents fully the Post-Architectural Model.

This implemented tool provides a range on its cost, effort and schedule estimates, from best to worst case outcomes. It also allows the planner to easily act on "what if" scenarios by quickly performing effect adjusting requirement, resources and staffing might have on estimated costs and schedules e.g. for risk management or job bidding purposes. The software devotion of the model also follows a naming assembly. The first release by USC was USC-COCOMO II.1997.0, which was then followed by USC COCOMO II.1997.1, with the current tool being called USC COCOMO II.1998.0. The calendar year at the end identifies the calibration. COCOMO II has more cost drivers (7 to 12 depending on sub module) than the earlier version.

6. COCOMO in Theory-W framework

The Spiral Model is a risk-driven process, dependent upon prototyping to map out the direction and extent of a project's development. In computing terms, the Spiral Model allows the developers to evaluate the software at each stage of its evolution, thereby keeping a constant check upon potential problems. More readily pinpointed and resolved at their source, problems unearthed in the Spiral process pose less of a cost threat in comparison to the back-tracking that would have to be done if a fault were to be discovered in the latter stages of software development using the Waterfall model. That is not to say that it does not demand considerable risk assessment expertise to ensure the project's success, which brings us to the relevance of COCOMO in the WinWin environment.

COCOMO complements the WinWin Spiral Model by providing a very useful tool: a cost estimation equation (Boehm et al. 1994). Once an option gets chosen, it gets refined in a collaborative manner. The refinements and their resulting tradeoffs are explored using detailed models. The current tradeoffs provided in WinWin-1 consists of an interactive Constructive Cost Model (COCOMO) tool for cost, schedule, functionality, performance trade-off analysis.

7. Some applications of the WinWin Model

An increasing number of companies (Boehm 1998) have started to use the Spiral Model as a development tool. EDCS (Evolutionary Design of Complex Systems) are engaged in a project, the aim of which is to use the WinWin Spiral Model in order to develop distributed collaborative negotiation aids and interactive analysis tools. One of the main tasks of the project is to refine and extend the WinWin System and WinWin Spiral Model. The WinWin Spiral Model has also been used experimentally by Aerospace Corp. and TRW in their implementation of satellite ground systems.

7.1. The WinWin Spiral Model—a case study

The particular Case Study (Boehm 1998) discussed in this paper involved delivering multimedia applications for the Integrated Library System of the University of Southern California. The system was designed to manage the acquisition, cataloguing, public access and circulation of library materials. Fifteen teams (six graduate students per team) were commissioned with the task of prototyping, planning, specifying and implementing 12 multimedia applications using the WinWin Spiral Model for software development. From the problem statements prepared by the library staff, the teams had to generate detailed specification in 11 weeks. The aim was to identify a feasible set of LCO's for a range of applications despite the following Constraints:

- Limited budget.
- Disruption to library services.
- Limited resources.
- Vague requirements.

The risks involved included control and communication difficulties. The difficulties were resolved by focusing each team on a single application area and giving them a common set of guidelines.

There were three primary stakeholders:
Developers (Software Engineering Students)
Customers (Librarians)
Users (Other Students)

Figure 4. The three primary stakeholders in the Integrated Library System.

The following illustrates these three primary stakeholders and their win conditions:

*7.1.1. **Library operations community.***

- Continuity of network operations and services
- Minimal disruption to service
- Career opportunities for systems administrators
- More efficient operations through technology

*7.1.2. **Library information technology community.***

- The Dean's vision that there should be transition to digital capabilities
- Evaluation of multimedia archiving and access tools
- Empowering library multimedia users
- Improvement of library staff digital capabilities
- Leveraging of limited budget for advanced capabilities

*7.1.3. **Centre for Software Engineering.***

- Similarity of projects for fairness and project management
- Reasonable match to the WinWin spiral model
- 15–20 projects (having 5-6 students per project team)
- Achieving a meaningful LCA in one semester
  Adequate network, computer and infrastructure resources

The WinWin Spiral Model was applied in 4 iterative cycles:

7.1.4. Cycle 0. Students determined the feasibility of the stakeholders Win conditions. Project guidelines were negotiated with other stakeholders.

7.1.5. Cycle 1 Life Cycle Objectives. LCO's, prototypes, plans and specifications were developed for each individual project. Reasons were given for wanting to do each project. It was shown to be possible to achieve the result in at least one way.

Each team of students worked with a library user and negotiated mutual agreements with a feasible architecture as determined by LCO. This was accomplished by using the WinWin GroupWare Support System for Requirements Negotiation. This provided WinWin forms for the stakeholders to express their Win conditions, define issues and deal with any possible conflicts. The roles of the three stakeholders were identified:

The operational concept and requirements team worked with possible users to report upon any possible user concerns such as software reliability, user-friendliness, fastness and flexibility.

The developers built upon problem statements given to them by librarians and users. These statements were vague, unrealistic and lacked detail. From these the students were expected to build up a consistent set of prototype plans and specifications.

Due to an underestimation of the training required and the time consumed when negotiating with a large number of stakeholders in such a short period of time it took the students longer than expected to implement the WinWin GroupWare.

Developers
The architecture and prototype team members were responsible for reporting developer concerns such as the use of familiar packages, support tools and any other technical challenges.
Customers
The plan and rationale team members were responsible for reporting the customer concerns of the librarians such as the limited time and budgets etc.
Users
The operational concept and requirements team worked with possible users to report upon any possible user concerns such as software reliability, user friendliness of software, fastness and flexibility of software.

Figure 5. The roles of the stakeholders.

7.1.6. Cycle 2 Life Cycle Architecture. Each team of students chose an LCA for their task and built upon their LCO. They encountered the following problems:

- Inconsistencies amongst artefacts.
- Failure to specify quality attributes.
- General misunderstanding about the applications scope.
- Inability to recognise that the plan was to focus on the development
- activities to come in cycle 3.

This resulted in delays, changes in prototyping equipment and the destabilisation of some of the WinWin Conditions. Furthermore, as soon as the other stakeholders saw the prototypes that had been engineered they wanted to change their requirements! However, all stakeholders obtained more realistic perceptions of what could be achieved.

7.1.7. Cycle 3 Initial Operational Capabilities. Due to a new semester only six projects were actually continued. Students were working on projects that they had not started. Only one project was actually implemented. It was the only application with sufficient budget, staff and sustainable facilities. Despite this being the only project to deliver the IOC in the long term all projects were successful in the short term. A major reason for the success was the strong emphasis on risk management. The general outcome was that all of the projects were successful in meeting the stakeholders Win conditions.

8. Conclusions

The WinWin Spiral Model provides discipline, flexibility and helps keep the project on schedule. If the variations are applied separately to a project the outcome will generally be unsuccessful, producing separate systems with many unnecessary and conflicting components. It is often the case that designers look beyond the parameters of the system objectives.

It is very difficult to get an assorted group of stakeholders to agree in a large project, due to team mix; this is why the incremental and evolutionary acquisition is important and referenced in the milestones. GroupWare was an important hurdle to cross initially; however, internet WinWin negotiation has made this problem obsolete. System requirements often need to be negotiated or new ones added on and there has been a problem in modifying groups of artefacts, when halfway down the process.

Many designers agree that part of the solution is expectations management; therefore, COCOMO II is used to help all concerned calibrate their expectations to what is achievable in the schedule. Due to software development trends such as reuse and COTS packages, COCOMO II will always require concurrent approaches to estimation.

It is a well known problem in large software developments that the user issues are submerged by financial trade off and licensing concerns, even though these are key concerns in the WinWin Model. Until recently, WinWin has not been cost effective enough to use in small projects and with the downward pressure on systems development costs this makes cost benefit calculations even more important than before if the spiral model is to grow. The future of the Spiral Model possibly lies in cost estimation of web page development, as this is a relatively untouched area or in electronic commerce as this industry is in its infancy.

References

Boehm, B. 1988. A spiral model of software development and enhancement, *Computer* May, pp. 61–72.

Boehm, B. 1995. Anchoring the software process.

Boehm, B. 1998. WinWin Spiral Model: A case study, *IEEE.*

Boehm, B. et al. 1994. Software requirements as negotiated Win conditions. IEEE Proc., 1994 Int. Conf. Requirements Eng., April 1994.

Boehm, B., Bose, P., and Horowitz E. et al. 1994. Software requirements as negotiation and regeneration aids: A Theory-W based approach, IEEE Proc. 17th ICSE, October 1994, pp. 1–12.

Boehm, B., Egyed, Kwan, Port, et al. 1998. Using the WinWin Spiral Model: A case study, *Comput. Weekly* July, p. 34.

Bose P. 1995. A Model for decision maintenance in the WinWin collaboration framework. 10th KBSE Conf., November, pp. 3–6.

Horowitz, E. 1999. *WinWin Reference Manual*, p. 5.

online: *http: // sunset.usc.edu / classes / classes.htm*, November 17, 1998.

online: *http: // sunset.usc.edu / cocomoII / cocomo.html*, 1998.

Pressman, R. S. 1998. *Software Engineering, A Practitioner's Approach*, New York, McGraw-Hill, pp. 42–45.

Royce, W. W. 1970. Managing the development of large software systems: concepts and techniques. Proc. WESCON, August 1970.

Sommerville I. 1997. *Software Engineering*, Reading, MA, Addison-Wesley.

Sommerville, Sawyer. 1997. *Requirements Engineering—A Good Practice Guide*, New York, John Wiley, pp. 54–61.

Peter Oriogun is currently a Senior Lecturer in Computing at the University of North London, England. Before joining the UNL, he worked as a Systems Programmer for Rolls Royce plc (Aero Engine Division), a Lecturer in Software Engineering Methods at Sheffield Hallam University, a Lecturer in Computing at Newham Community College, a Lecturer in Information Technology at South Bristol College and for Community Service Volunteers as a Computer Consultant. His research interest is in Teaching and Learning of Computing in Higher Education, and software metrics.

Chapter 13

TOWARDS STUDENTS' NEGOTIATED INCREMENTAL MODEL -NIM: A CASE STUDY ON STUDENTS' IN-COURSE-ASSESSMENT IN A SOFTWARE ENGINEERING MODULE

Peter K Oriogun, University of North London

School of Informatics and Multimedia Technology, University of North London,
London N7 8EA
Tel +44(0) 207 973 4852, fax +44(0) 207 753 7009
Email: p.oriogun@unl.ac.uk , http://www.unl.ac.uk/simt/staff.shtml
KEYWORDS:
WinWin Spiral Model, Rapid Application Development (RAD), Incremental Development, Capability Curriculum, Spiral Model, Negotiated Incremental Model (NIM).

ABSTRACT

This paper reports a study designed to show the effect of adopting iterative life cycle process models in order to justify the adoption of the WinWin Spiral model as a useful process model to use for students' In-Course-Assessment within a semester framework for a software engineering module. A case study using the WinWin Spiral process model for the development of the In-Course-Assessment for the module, Software Engineering for Computer Science is presented. The case study looks at how a particular group of students implemented the WinWin decision support framework (Boehm, 1996a) in the resolution of conflicts through negotiation after the stakeholders have had the opportunity to raise issues, select options and finally arriving at a negotiated agreements satisfying all the parties involved. A statistical analysis of the outcome of the practical aspect of the module Software Engineering for Computer Science for is presented for the academic session 1999/2000. It shows the statistical models supporting the iterative life cycle process models adopted for the development of the School of Informatics and Multimedia Technology (SIMT) Golf League. The paper proposes a students' iterative life cycle process model, Negotiated Incremental Model -NIM. The NIM process model will be able to deal with the establishment and adherence to team responsibilities, close co-operation between team members, with good established methods and means of communication in the development of software within a semester framework.

1. INTRODUCTION

This paper reports a case study designed to show the effect of adopting different prototyping life cycle models on students' practical In-Course-Assignment (ICA) in a Software Engineering module during the academic session 1999/2000. The main objective was to establish (if possible) an ideal software life cycle model within a one semester framework for students on three different courses at the University of North London (UNL). This study is also a follow-on to a previous paper (Oriogun, 2000a) that recommended addressing the shortcomings of WinWin Spiral Model (Boehm, 1996b) by providing case studies with different iterative software life-cycle models in order to justify the usefulness and/or effectiveness of the same. Four software life cycle models were used for the study, namely, the WinWin Spiral Model, Rapid Applications Development -RAD (Kerr and Hunter, 1994), Spiral Model (Boehm, 1988) and the Incremental Development Model (McDermid and Rook, 1993). The case study presented in this paper scored the highest mark overall in comparison to the rest of the life cycle process models adopted for the development of the software for the module, Software Engineering for Computer Science during the academic session 1999/200. The paper also looks at how the students dealt with conflict identification, negotiation and resolution in the development of the SIMT Golf League System over two spiral cycles, and proposes the students' iterative life cycle model, the Negotiated Incremental Model -NIM method.

2. BACKGROUND TO THE SOFTWARE ENGINEERING MODULE-IM283

The module Software Engineering for Computer Science (IM283 Home Page) was developed in 1998 (Oriogun, 2000b) to incorporate the Capability Curriculum (Capability Curriculum Page). The Capability Curriculum is a new invention of University of North London, the aim of which is to provide graduates with all-round capability to industry and commerce. The aim of the module is to enable students to understand, participate in, control and manage the process of developing large complex software product. It is intended to enhance students ability to program effectively in at least one programming language, together with the necessary data modelling techniques to analyse, design and implement a detailed software project as part of a team. The recommended text (Pressman, 1998) is a detailed textbook on software engineering, it covers more topics than can be possibly be covered within one semester.

2.1 IM283 TEACHING AND LEARNING METHODS

During the lecture periods, students were introduced to the module, exposed to different software life-cycle paradigms and project management techniques (10 hours). The importance of group-work was an on-going theme throughout the module. Tutors advised on the weekly deliverables for the module; they also monitored, amended and/or corrected the intermediate deliverables submitted by students allocated to their tutor group throughout the semester (24 hours). Software quality issues were a subject of two lectures (4 hours) and the various types of software testing strategies were the subject of one lecture (2 hours). Students were expected to spend some time (72 hours) unsupervised for their group coursework and also some unsupervised time for individual coursework (24 hours). The pre-requisites for IM283 are a pass in Further

Visual Basic Programming or C++ Programming or Rapid Application Development and a completion in the Systems Analysis and Design module. There are six Learning Outcomes -LOs (IM283 Home Page) for this module. The group ICA maps three of the LOs to one of the two capabilities being developed in this module, namely, to "Manage self and relate to others -Capability C4 (Capability Curriculum Page).

2.2 IM283 TEACHING ARRANGEMENTS FOR THE GROUP IN-COURSE-ASSESSMENT (ICA)

This module has a module convenor (the person responsible for the delivery and assessment of the module) and four tutorial assistants. Tutorial Assistants monitor the weekly deliverables on the group In-Course-Assessment (ICA) for the module, and provide useful feedback to students regarding their progress. Each tutorial group has a maximum of six coursework groups, with each coursework group having a maximum of six students. The class size is typically 120-160. This figure includes those students registered for the module during 1999/2000 academic session and those being reassessed in examination or ICA or both based on their tuition from 1998/1999 academic session. All modules taught at the University on the modular degree scheme are fixed, weekly, four-hour blocks. The four hours are split into a two-hour lecture and two-hour tutorials. The tutorial focuses mainly on the group coursework assignment for the module. The coursework element of the module, relied heavily on students having a certain degree of knowledge and understanding of how to synthesise and analyse a software engineering problem, using a combination of paradigms and the mapping of that analysis onto a design.

3. REQUIREMENTS SPECIFICATION FOR THE GROUP ICA

The specification of a fictitious SIMT Golf League was issued to students in the first week of the semester. Each group of students had the freedom to decide on which of the four given prototyping models they will adopt in the development of the required software from the specification given. Each tutorial group must have at least four different groups, each of the groups adopting either the Theory-W Based Spiral Model (also know as the WinWin Spiral Model) or Rapid Application Model or the Spiral Model or the Incremental Development Method. Replication is allowed within each tutorial group thereafter. The specification issued to students at the beginning of the module is described below.

"The members of the SIMT Golf League regularly compete in matches to determine their comparative ability. A match is played between two golfers; each match either has a winner and a loser, or is declared a tie. Each match consists of a round of 18 holes with a score kept for each hole. The outcome of a match is used to update the ranking of players in the league. The winner is declared better than the loser and any golfers previously beaten by the loser. Other comparative rankings are left unchanged. The software should keep information about each golfer, e.g. name, club ID, address, the date of last golf match, and current match ranking etc. Each round of golf should also be tracked. The software should allow golfers to input their own scores and allow any legal user to query any information in the system."

4. PROTOTYPING AND THE ITERATIVE LIFE CYCLES ADOPTED FOR IM283 GROUP ICA

This section gives a brief introduction to prototyping in general, with particular emphasis on the adopted prototyping methods for IM283 group ICA, namely, the Rapid Application Development (RAD) method, the Spiral Models and the Incremental Development model.

4.1 PROTOTYPING

The use of prototypes (Salis, Tate and MacDonell, 1995) has reduced the level of frustration in the systems development world and has improved systems quality and productivity because it has increased systems acceptability. Furthermore, it has helped the concept of real user participation in the systems development process. The effect of prototyping on requirements specification within the software process development for non-trivial applications is an ever-growing and ever-changing set of user needs and expectations (Davis 1993). Another author, (Case, 1986) suggests that prototyping as a technique is language independent, and regards prototypes as being functional information systems but should only be regarded functional and part-functional models of information systems that are built during the development process. Furthermore, (Lantz 1986), explains the rational for choosing the prototyping approach to systems development, and addresses the issues and relative advantages of developing mock-up prototypes, compared with full prototypes which include some conventional processing.

4.2 THE RAPID APPLICATION DEVELOPMENT (RAD) MODEL

The RAD model (Pressman, 2000) is a "high speed" adaptation of the linear sequential model. It refers to a development a development lifecycle designed to give much faster development and high quality results than the traditional lifecycles (Martin, 1991). The RAD approach is depicted in Figure 1 (Kerr and Hunter, 1994):

- **Business Modelling:** this phase is described as the information flow amongst business functions.
- **Data Modelling:** this phase is the information flow (from business modelling) refined to support the business.
- **Process Modelling:** this is the transformation of data object (from the data modelling phase) to achieve the information flow needed to implement a business function.
- **Application Generator:** this is the use of 4th Generation Languages (4GLs), instead of creating software using third generation programming languages. As such it makes use of existing programs and/or libraries where necessary. CASE tools are used to automate construction of software.
- **Test & Turnover:** this is the testing of newly acquired programs and the reuse of existing ones. Less time will be spent on testing based on the fact that we already know the behaviour of existing software.

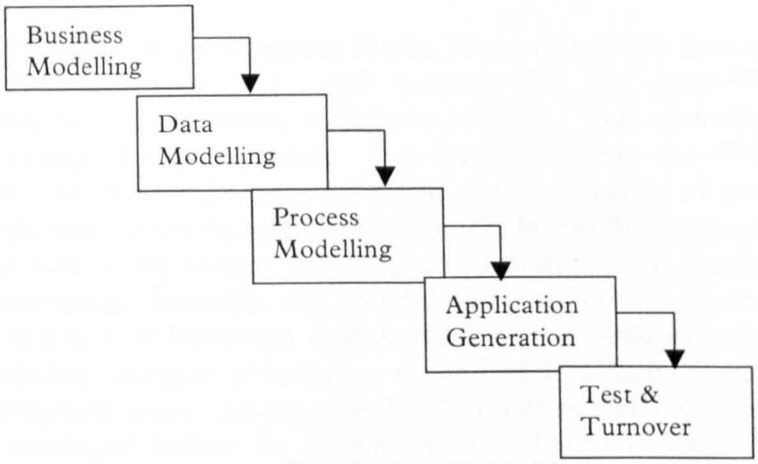


Fig. 1: The RAD Model

4.3 THE SPIRAL MODELS

The WinWin Spiral Model (Boehm, 1996a and Oriogun, 1999) extends the original Spiral Model by adding Theory-W activities to the front of each cycle. The theory argues that a project will only succeed if the critical stakeholders are all winners.

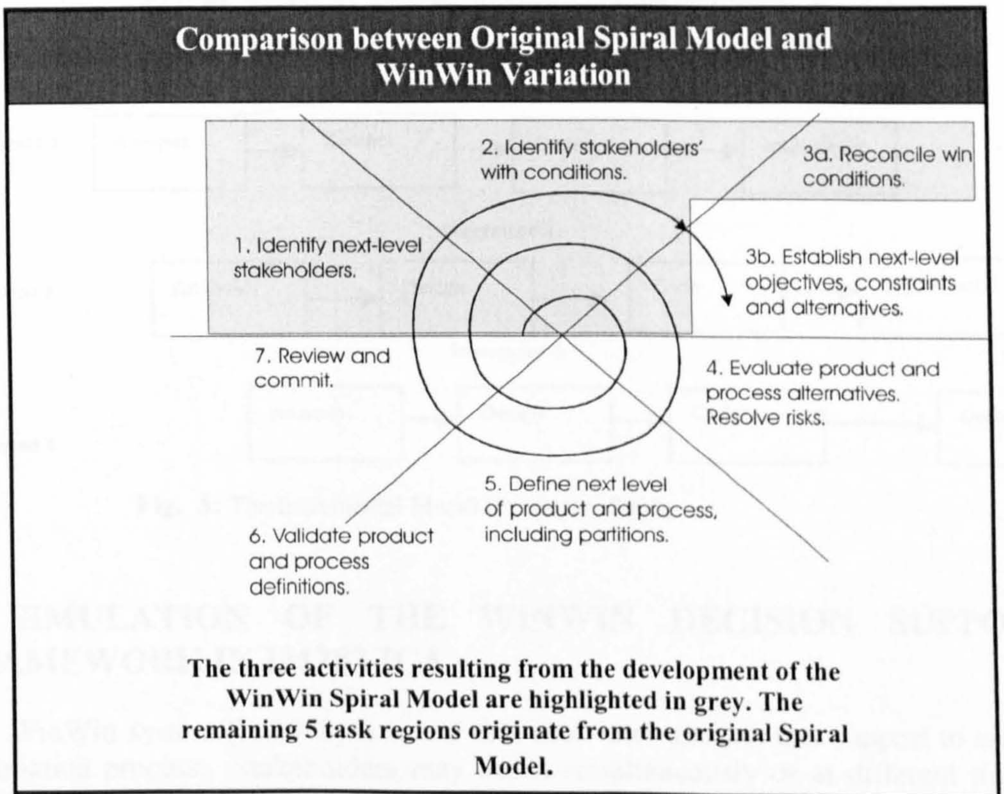


Fig. 2: The WinWin Spiral Model (Boehm, 1998)

4.4 THE INCREMENTAL DEVELOPMENT MODEL

The Incremental Development Model (Pressman 2000) focuses on the delivery of an operational product with each increment. The emphasis of the incremental development is to develop as little as possible. This normally equates to developing just enough for an increment. This model combines the elements of the classic life cycle (the Waterfall) model with the iterative nature of prototyping. Incremental Development *prioritising* and *scheduling* falls into three categories. Firstly implement those part of the system that must be in place before functional increments can be implemented. Secondly, one could develop the critical increments first by providing the interface or functional threads, which are needed as soon as possible. The last scheduling category is based on a ratio of two factors, namely, user benefits and development costs. An increment with a high perceived benefits and high costs, may be developed before an increment with low cost and low perceived benefits. Alternatively, increments that have high benefits and low cost will more often than not be developed earlier. This method produces a series of mini life cycles rather than the construction of one monolithic life cycle. Each linear sequence (below) produces a deliverable, an *increment* (MacDermid and Rook, 1993).

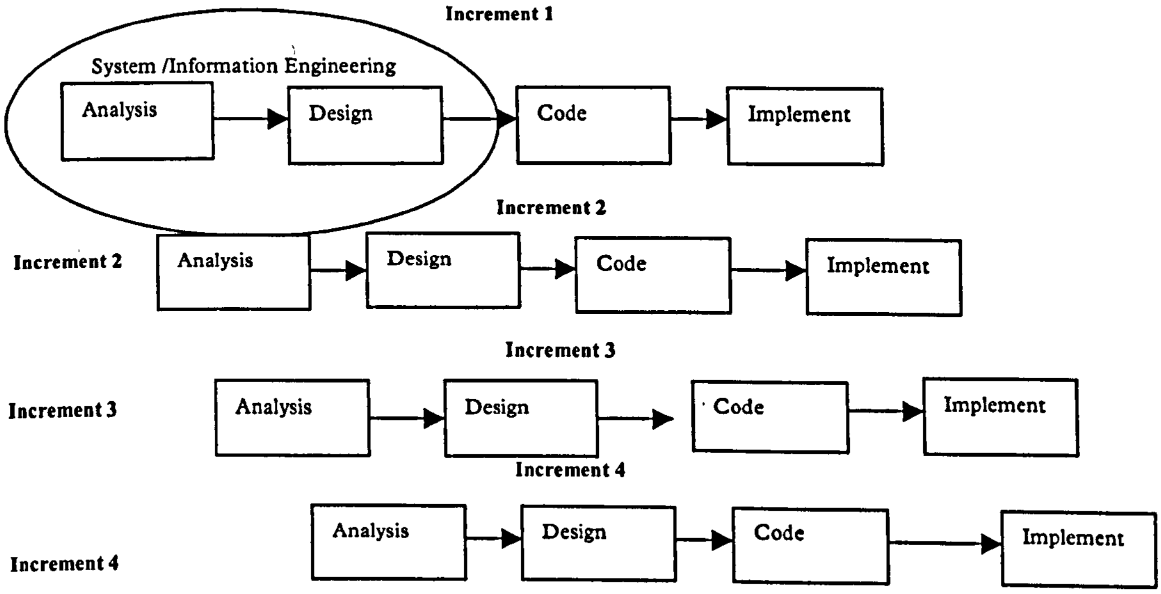


Fig. 3: The Incremental Model (Pressman, 2000)

5. SIMULATION OF THE WINWIN DECISION SUPPORT FRAMEWORK IN IM283 ICA

The WinWin system (In,1998) is a tool that uses inter and intranet support to aid the negotiation process. Stakeholders may use it simultaneously or at different time to negotiate. A number of support tools may be used with the WinWin system to assist in the negotiation (QARCC - Quality Attribute and Conflict Consultant, COCOMO - COnstructive COst MOdel etc.). In order to model the WinWin system, a news conferencing facility was set up using eGroups (eGroups Web Page) and WebCT. Each group of students (particularly those using the WinWin Spiral Model) were asked to set up and maintain their own conferencing facility in order to negotiate. However, students had to use other means (face-to-face) to fully explore the type of facilities available within the WinWin system. Figure 4 shows the WinWin architecture. The module convenor and the tutorial assistants acted as the 'Systems Engineer' as well as the 'User', whilst the students acted as 'Developer and 'Customer'.

The WinWin support framework is a Unix workstation based computer program that aids the capture, negotiation and co-ordination of requirements for a large system. It assumes that a group of people called the stakeholders, have signed on with the express purpose of discussing and refining the requirements of their proposed system.

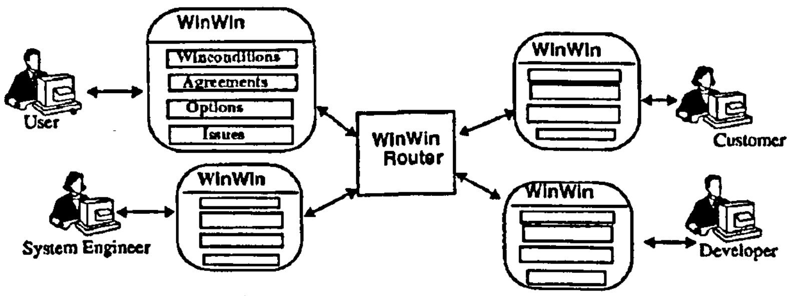


Fig. 4: The WinWin Architecture (Boehm 1998 and Oriogun 1999)

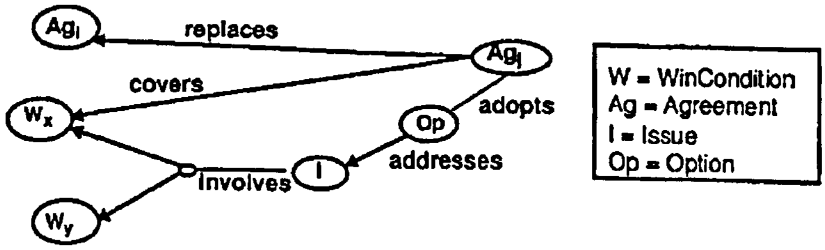
5.1 WINWIN DECISION MODEL

The WinWin process is modelled using four main objectives (Boehm,1988):

- Win Condition
- Issue
- Option
- Agreement

The reconciliation phase attempts to resolve conflicts between win conditions. If a win condition is non-controversial (there is no conflict), it is covered by an agreement (Ag). Relationship between win conditions are established, leading to issues (I) being identified which raise the conflicts between win conditions and their associated risks and uncertainties. Options (Op) are considered which suggest strategies for resolving issues which lead to agreements (Ag) that satisfies stakeholders win conditions and also define the systems objectives.

Fig. 5: WinWin decision objects and relations between them (Boehm,1988)



6. WINWIN FORMS SPIRAL CYCLE 1 - A CASE STUDY FROM IM283 ICA

The case study that generated the WinWin conditions in Tables 1-6 below equates to two rounds of negotiations. This was due to the fact that the project had to be completed within a semester framework. Tables 1-3 forms the basis of the WinWin conditions of the stakeholders, the WinWin Conflicts and Issues and the WinWin negotiated agreements for the first spiral cycle. Tables 4-6, show the second phase of negotiations. The implemented software thus relates to this final round of negotiations.

The particular group used for the WinWin case study herein, scored the highest marks for their efforts in the research, analysis, design, development and implementation of a viable software system using Visual Basic and Access database.

Stakeholders	Win Condition
Customer / Administrator	1. The project remains within budget. 2. The project is delivered within time. 3. The software is reliable. 4. The software enables non-technical people to manipulate it. 5. The storage space required for data is kept to a minimum. 6. The software does not allow cheating 7. The software enables the administrator to input, update and delete golfer details. 8. The software enables administrator to input data relating to matches. 9. The system has password access. 10. The club's ranking system is automated. 11. Matches will be scored automatically 12. The system provides enhanced security. 13. Information will be easy to retrieve. 14. Data integrity will be maintained. 15. Access to sensitive information will be restricted.
User/Golfer	1. The system is easy to manipulate. 2. Paper store cards are dispensed with. 3. Matches will be scored automatically. 4. The ranking system is logical. 5. Internet access to system. 6. Past matches score cards are available. 7. The system allows golfers to browse the record of other golfers. 8. The system will be easy to learn and not too technical.
Development Team	1. The project is delivered on time. 2. The project is delivered within budget. 3. The software satisfies administrator and the golfer's requirements. 4. The team gains valuable skills and experience.

Table 1: WinWin Conditions of Stakeholders -Spiral Cycle 1

Table 2 below shows the Issues, Conflicts and the Options between stakeholders for the IM283 ICA Golf League System for the group selected for this case study.

Issue/Conflict	Between which Parties	Options
Nature of system Is it to be on-line? On-line system would fall out of budget and could not be delivered within the time allowed.	Developer / Customer / User	1. Implement the system on-line and go over budget and over time. 2. Build the system on a single PC and develop the on-line capability at a later date when funds become available. Use a language and implementation which would allow web access at a later date.
Scorecards The user would like the complete removal of paper score cards from the system. The only way to do this is to provide hand-held PCs. The administrator does not want to go out of budget.	Administrator/ User	1. Buy hand-held and go out of budget and over time. 2. Retain the paper score cards. 3. Configure the system so that additional functionality such as hand-held PC scoring can be adumbrated at a later date.
Security	Customer / User	The customer wants to keep golfer records hidden from the user. The user, however, wants to view the details of his competitors.
Rankings	Customer / User	The administrator preferred the old ranking system. The users rejected the old system because it is too unrealistic.

Table 2: WinWin Issues / Conflicts / Options -Spiral Cycle 1

Table 3 below shows the negotiated Agreements between stakeholders for the IM283 ICA Golf League System for the group selected for this case study.

Issue	Parties	Agreement
On-line system	Administrator / Developer/ User	Develop system using Visual Basic, MS Access. This will allow the system to be on-line once additional funds have become available.
Score Cards	Administrator / User	Retain paper score cards initially. The technology used will allow for an easy conversion to hand-held scoring once the funds for such technology become available.
Scoring System	Developer / User / Customer	The old scoring system is unworkable and will be replaced with a more realistic one. In the new system, a golfer will get 3 points for a win, 1 point for a tie and 0 points for losing.
Security	User / Customer	The new system should provide both a user interface and an administrator interface. The administrator will be able to view all information and the user will have restricted access to other golfer's details.

Table 3: WinWin Negotiated Agreements -Spiral Cycle 1

Table 4 shows the second spiral cycle of the WinWin negotiation between stakeholders for the IM283 ICA Golf League System for the group selected for this case study.

Stakeholders	Win cndition
Customer / Administrator	The system remains within budget. The system remains on schedule The system provides search functions The system tracks subscriptions The system interface looks like a traditional computer system.
User / Golfer	Additional documentation is added in the form of a tutorial. The interface is developed in a style which reflects a golf club.
Development Team	The system remains within budget. The system remains on schedule. The team receives good marks for the project. The project is 'do-able'. The team gains valuable experience in software engineering.

Table 4: WinWin Conditions of stakeholders -Spiral Cycle 2

Table 5 below shows the issues and conflicts within the second spiral cycle for the IM283 ICA Golf League System for the group selected for this case study.

Issue / Conflict	Between which Parties	Options
Interface Design	User / Customer	No frills interface. Interface with more graphics colour etc.
Help Facilities / Budget / Project Schedule	User / Customer / Developer	Implement limited Help facility. Implement full help facilities, including help tutorial and documentation.

Table 5: WinWin Conflicts / Issues -Spiral Cycle 2

Table 6 below shows the final negotiated agreements reached at the end of spiral cycle 2 for the IM283 ICA Golf League System for the group selected for this case study.

Issue	Parties	Agreement
Interface Design	User / Customer	As the graphics and colour will not add to the cost, the system should be designed to reflect the context in which it will operate.
Help Facilities / Budget / Project Schedule	User / Customer / Developer	In order to stop the project going over budget, a full help tutorial will not be implemented. However, the Help menu will contain a 'How To' section.

Table 6: WinWin Negotiated Agreements -Spiral Cycle 2

7. ANALYSIS OF THE IM283 ICA RESULTS (1999/2000 ACADEMIC SESSION)

The main purpose of this study is to compare the WinWin Spiral model to other prototyping life cycle models for students' group ICA in order to measure its usefulness and/or effectiveness within a semester framework. The detailed statistics that follows shows that there was no significant difference between the final results of those students adopting the WinWin Spiral model (26 in total) and those adopting the Incremental Development method (4 in total). The WinWin Spiral model and the Incremental Development method ICA results are shown in Table 7, given one first year prerequisite (Introduction to Visual Basic Programming -IM102) and one second year prerequisite (Systems Analysis -IM201). Tables 7-9 shows that 39% of the students adopted the WinWin Spiral model, 6% adopted the Incremental Development method whilst 33% adopted the original Spiral model (Boehm, 1988) and 22% adopted the Rapid Application Development (Martins, 1991) method.

7.1 EXCLUDED STUDENTS FROM THE GROUP PRACTICAL ICA STATISTICS

In order to treat each student as having the same prior knowledge before registering for IM283, the module convenor established the results of those students who had registered on, and completed the "Introduction to Programming -IM102" module. A further prerequisite is a completion of "Systems Analysis -IM201" module. This is a first semester module in the second year, and it is core for the three courses. Module codes are available, through a link (Undergraduate Computing Syllabi) to the School of Informatics and Multimedia Technology teaching page. Students transferring onto the degree programme from the Higher National Diploma (HND), either at the end of the first semester of the first year, or at the end of the second semester of the first year of their HND programme, are excluded from the statistical analysis of results that follows. Therefore, from a population of 98 students who completed the group ICA, a valid sample size for the study totalled 67 students. Tables 7-9 below shows the valid sample size of students adopting the prototyping methods in developing their software for the SIMT Golf League (IM283 Home Page), including their ICA results for IM283 (all the marks are in percentage).

WinWin / Incremental			
Stud No	IM 102	IM 201	IM 283
1	61	67	56
2	35	68	56
3	69	72	76
4	56	66	65
5	41	33	45
6	68	63	70
7	78	66	70
8	78	77	70
9	80	68	70
10	56	62	56
11	70	61	56
12	67	67	56
13	61	64	56
14	30	54	63
15	33	63	42
16	55	70	50
17	67	84	84
18	77	86	84
19	67	45	37
20	42	61	37
21	48	29	45
22	54	71	63
23	64	26	58
24	57	66	58
25	64	65	58
26	43	79	58
27	40	58	73
28	53	54	73
29	45	50	73
30	65	64	54

Table 7: WinWin /Incremental

Spiral			
Stud No	IM 102	IM 201	IM 283
1	28	57	55
2	64	53	32
3	57	69	42
4	26	46	45
5	48	60	45
6	49	63	45
7	77	39	45
8	51	62	55
9	37	69	56
10	54	69	52
11	53	60	35
12	70	46	35
13	73	64	38
14	54	23	38
15	60	61	38
16	41	80	58
17	54	79	66
18	43	43	58
19	56	76	42
20	46	62	43
21	60	37	43
22	59	76	42

Table 8: Spiral

RAD			
Stud No	IM 102	IM 201	IM 283
1	83	63	74
2	45	69	73
3	64	54	71
4	56	59	22
5	45	67	28
6	60	69	28
7	56	70	63
8	27	65	63
9	51	63	63
10	89	60	63
11	26	65	38
12	27	62	38
13	54	27	28
14	54	23	25
15	50	60	85

Table 9: RAD

7.2 STATISTICAL MODELS SUPPORTING THE OUTCOME OF THE GROUP IM283 ICA

A detailed analysis of the statistics revealed that the module "Introduction to Programming" (IM102) does not have any effect on students' results, it is therefore not significant, and was removed from the two models. The module IM201 is significant (Oriogun, 2000b) as a prerequisite for the IM283 ICA. Two forms of statistical models were considered using the GLIM statistical package (GLIM, 19993), namely a *variate-factor interaction* model and a simpler model which forces influential data points to be graphed with the same common slopes, and different intercepts (see tables 7-9 for the data points involved).

7.21 THE INTERACTION MODELS

IM283 ICA	= 40.87	+0.3204 *IM201	if WinWin / Incremental
	= 40.87 - 7.6	+0.218 *IM201	if RAD
	= 40.87 -32.6	+0.810 *IM201	if Spiral

From the data in Tables 7-9, there is no strong evidence (p=0.06) that this more complicated (interaction) model significantly explains more of the variation in the data. We therefore preferred the simpler model below (see Figure 6 for the diagram) without the variate-factor interaction.

7.22 THE NONE INTERACTION MODELS

IM283 ICA	= 39.94		+ 0.335 *IM201	if WinWin/Incremental
	= 39.94 -13.86		+ 0.335 *IM201	if RAD
	= 39.94	-11.14	+ 0.335 *IM201	if Spiral

8. DISCUSSION OF STATISTICAL MODELS SUPPORTING THE LIFE CYCLE MODELS

Student groups could choose which prototyping model they adopted (subject to each model being covered within a tutorial group), there is therefore a possibility of bias, which could have been avoided by randomly assigning methods to student groups. Weaker groups could have chosen what they see as easier methods, or better groups might choose the WinWin model because they feel it will gain them more marks. A neutral marking scheme was adopted between the methods, as such it is hoped that the choice of method will not mitigate against any of the groups. . The non-variate interaction model (that follows) is a simpler model which forces influential data points to be graphed with the same common slopes, and different intercepts. This section interprets the statistical models supporting each life cycle model adopted for the module Software Engineering for Computer Science In-Course-Assessment.

8.1 THE WINWIN SPIRAL PROCESS MODEL / INCREMENTAL DEVELOPMENT METHOD

The statistical models described in section 7.22 are the best (and simple) models explaining the outcome of each prototyping model adopted for the module IM283 ICA (Software Engineering for Computer Science In-Course-Assessment). The statistical models for those students adopting the WinWin Spiral Process Model or the Incremental Development Method (see Table 7) are almost the same. A better

approach therefore treats these groups of students as one and compares them with the Spiral and the RAD. The regression parameters slightly changed by the grouping and give:
IM283 ICA = 39.94 + 0.335 * IM201

This suggest that students could score around a minimum of 40% for the module Software Engineering for Computer Science In-Course-Assessment (IM283 ICA) without having completed the prerequisite module, Systems Analysis and Design if they use the WinWin Spiral Process Model, or the Incremental Development Method. In other words the students using the WinWin Spiral Process Model and those using the Incremental Development Method could score in the range (40%,73%).

8.2 THE RAPID APPLICATION DEVELOPMENT METHOD (RAD)

The statistical model supporting those students using the Rapid Application Development (RAD) method is as follow:
IM283 ICA = 39.94 -13.86 + 0.335 * IM201

This suggest that students could possibly score around a minimum of 26% for the module Software Engineering for Computer Science In-Course-Assessment (IM283 ICA) without having completed the prerequisite module, Systems Analysis and Design, if they use the RAD method. In other words those students using the RAD method could score in the range (26%,60%).

8.3 THE SPIRAL MODEL

The statistical model supporting those students using the Spiral model is as follow:
IM283 ICA = 39.94 -11.14 +0.335 * IM201

This suggest that students could possibly score around a minimum of 29% for the module Software Engineering for Computer Science In-Course-Assessment (IM283 ICA) without having completed the prerequisite module, Systems Analysis and Design, if they use the Spiral model. In other words those students using the Spiral model could score in the range (29%,62%). There is strong evidence to suggest (from the statistical models supporting the iterative life cycle models adopted) that within a semester framework, students will probably find it easier and beneficial to develop a software from inception to completion using the WinWin Spiral Process Model or the Incremental Development Method.

9. CONCLUSIONS AND FUTURE WORK

In this paper, the author investigated the relative merits of adopting different iterative life cycle process models for developing and implementing software in a Software Engineering module, using teams of students within a semester framework. The investigation has led the author to believe that it is possible to develop a process model, similar in some parts to the WinWin Spiral model, and, in other parts to the Incremental Development Method. It is not possible in a semester to fully explore other aspects of the WinWin Spiral model, such as risk analysis, COCOMO II, COTS etc. It is however, possible to deliver increments of the functional and non-functional requirements from a specification by developing feasible aspects of the WinWin Spiral incrementally.

The author is proposing the students' iterative model, the *Negotiated Incremental Model* -NIM. The proposed students' iterative process model, NIM, will be able to deal with the establishment and adherence to team responsibilities, close co-operation between team members, good trusted methods and means of communication in the development of a successfully planned, organised and executed project with established guiding principles.

The NIM method is to be further researched and developed by the Software Engineering Research team at the University of North London. The NIM method will be able to operate effectively under a modular degree framework, as it will require specific prior studies before students can benefit from its use.

ACKNOWLEDGEMENTS

The author would like to take this opportunity to thank the following staff members and students for their contribution to this paper:

Professor Robert Gilchrist, Stella Odysseos, John Ndeta, Naomi Barbor, Adam Saltiel Gordon Mcintyre, Samer Harmood and Alireza Nabijou.

REFERENCES

Boehm, B W.(1988). A Spiral Model of Software Development and Enhancement, *Computer*, vol.21, no 5, May 1998, pp 61-72.

Boehm B W et. al., (1993). Knowledge Summary of the Focused Workshop: Next Generation Software Processes and Their Support, USC Centre for Software Engineering, October 27-29 1993, Edited by Mingjune Lee and Barry Boehm.

Boehm, B W.(1996a). Software Requirements As Negotiated Win Conditions, Proceedings of the *1st International Conference on Requirements Engineering*, ICRE96, April 1996.

Boehm, B W. (1996b). Anchoring the Software Process, *IEEE Software*, vol. 13, no. 4, July 1996, pp73-82.

Boehm, B W. (1998). Using the WINWIN Spiral Model: A Case Study, *Computer*, vol. 31, no. 7, July 1998, pp 33-44

Butler, J. (1994). Rapid Application Development in Action, Managing Systems Development, *Applied Computer Research*, vol. 14, no.5, May 1994, pp 6-8

Capability Curriculum Page. http://www2.unl.ac.uk/~head/capabil.htm.

Case, A. F. Jr (1986). *Information Systems Development*, Prentice-Hall, Englewood Cliffs, NJ.

Davis, A. M. (1993). *Software Requirements:Objects, Functions and States.* PTR Prentice-Hall, Englewood Cliffs, NJ.

EGroups Web Page.(2000). http://www.egroups.com, July 2000

In, H.(1998). Conflict Identification and Resolution for Software Quality Attribute Requirements, *PhD thesis*, University of Southern California, 1998.

IM283 Home Page. http://www.unl.ac.uk/simt/im283/

Kamm, R et al. (2000). Practice in IS Education: Live Projects for Livelier Students, Proceedings. *5th UKAIS Conference*, UWIC, 26-28 April 2000, Wales, UK, pp 349-355, ISBN: 0077097556, McGraw Hill.

Kerr, J and R Hunter. (1994). *Inside RAD*, McGraw-Hill, 1994.

Lantz, K. E. (1986). *The Prototyping Methodology*, Prentice-Hall, Englewood Cliffs, NJ.

Martin, J. (1991). *Rapid Application Development*, Prentice-Hall, 1991.

McDermid, J. and P Rook. (1993). Software Development Process Models, *Software Engineer's Reference Book*,CRC Press, 1993, pp 15/26-15/28.

Oriogun, P K.(1999). A Survey of Boehm's Work on the Spiral Models and COCOMO II - Towards Software Development Process Improvement, *Software Quality Journal*, vol. 8,51-60 (1999), Kluwer Academic Publishers, ISSN 09639314.

Oriogun, P K.(2000a). A Review of Theory-W Based Spiral Process Model and its Associated Tools for Supporting Conflict Identification and Negotiation in Software Development, Proceedings, *Software Quality Management VIII, Approaches to Quality Management*, April 2000, Greenwich, UK, pp 53-64, ISBN: 1902505255.

Oriogun, P K and R Gilchrist.(2000b). The Effect of Studying an Information Systems Module on Students' Results in a Software Engineering Module, Proceedings of the, *5th UKAIS Conference*, UWIC, 26-28 April 2000, Wales, UK, pp 394-401, ISBN: 0077097556, McGraw-Hill.

Pressman, R S. (1998). Software Engineering -A Practitioner's Approach, 4th Edition, European Adaptation, McGraw Hill 1998.

Pressman, R S. (2000). Software Engineering -A Practitioner's Approach, 5th Edition, European Adaptation, pp 37-38, McGraw Hill 2000.

Undergraduate Computing Syllabi.(1998). http://homepages.unl.ac.uk/~mumfordm/Revalidation/simtsyl.htm, School of Informatics and Multimedia Technology, University of North London,1998.

Faculty Undergraduate Award Scheme. (1998). http://homepages.unl.ac.uk/~mumfordm/Revalidation/simtdd.htm, School of Informatics and Multimedia Technology, University of North London,1998.

Salis, Tate and MacDonell. (1995). *Software Engineering: practice, managemet, improvement*, Addison-Wesley Publishing Company, 1995.

The GLIM 4 System. (1993). Release 4 Manual, Clarendon Press.

A LONGITUDINAL STUDY ON THE IMPACT OF INFORMATION SYSTEMS ANALYSIS AND DESIGN PREREQUISITE ON A SOFTWARE ENGINEERING MODULE

[1]Peter K Oriogun, [2]R Gilchrist
[1]School of Informatics and Multimedia Technology
[2]STORM: Statistics, OR and Probabilistic Methods
University of North London
[1]Tel: +0207 973 4852
[1]Fax: +0207 753 7009
Email: p.oriogun@unl.ac.uk
http://homepages.unl.ac.uk/~oriogunp

ABSTRACT

This is a follow-on paper to the one presented at the UKAIS 2000 (Oriogun and Gilchrist, 2000) reporting on the effect of studying an Information Systems module on students' results in a software engineering module. In this paper we investigate the impact of studying an Information Systems Analysis and Design module as a prerequisite for the module Software Engineering at the University of North London, from the cohort of students who started their modular degree programmes in Computing, Computer Science and Business Information Systems during the 1998/1999 academic session. These students will be graduating in the 2000/2001 academic session provided they have passed all the required modules for an award of a degree in one of the three titles mentioned herein. This paper is primarily concerned with the effect of coursework on students performance in a software engineering module; this module emphasises analysis, design and implementation of software products.

1. INTRODUCTION

This paper reports on a new module, Software Engineering for Computer Science, developed within the School of Informatics and Multimedia Technology during the1998/99 academic session incorporating the Capability Curriculum (Capability Curriculum, 2001) as prescribed by the University of North London. For the purpose of this paper we will refer to the module simply as Software Engineering (Software Engineering, 2001). The aim of the module is to enable students to understand, participate in, control and manage the process of developing a large complex software product. It is also intended to empower students in terms of their programming ability, team-work, and to raise their awareness with respect to the fact that software engineering is not just about writing code. As such, students were exposed to a variety of software life cycle process models, in order to appreciate that the process is just as important as the product (Oriogun, 2000). In this paper, we look at the statistical models supporting the performance of students on three courses, BSc (Hons) Computing, BSc (Hons) Computer Science and BSc (Hons) Business Information Systems, with respect to their coursework and overall unit marks in the module Software Engineering over the past three years. We do so firstly as three different courses and, secondly, as a whole group of students studying on the module Software Engineering, given that they must have a prerequisite of Information Systems Analysis and Design prior to registering on the module.

2. EXPLANATION OF REGRESSION LINE

In our study of the impact of Information Systems Analysis and Design (ISAD) prerequisite on a Software Engineering (SE) module we obtained a regression line of the following form.

$$SE = \alpha + \beta \, ISAD$$

Our equation above is saying that the expected (mean) SE is given by this relation. Note however that this is an estimated relation, with a sampling variability. We could estimate a confidence interval for this mean relation. A given student's mark will vary about the true mean value, with a variance which could be estimated. Thus, for example, if **ISAD** = 0, then **SE** = α is the average for such a student. However, a student with **ISAD** = 0 will not actually have **SE** = α. We can estimate the variance about the α, but the actual observation is of course unknown. Similarly, if **ISAD** = 100, **SE** = $\alpha + 100\beta$ is the average /expected score for such a student, but the actual mark will vary about the expectation. Again, we can estimate the variance about the line, although not the actual observation. In the following graphs, we show the 95% confidence interval for the expected SE score and the 95% prediction interval for any future SE score. The confidence interval gives us an indication of the variability of the expected score, i.e. if we repeat our samples, 95% of all such intervals would contain the true line. The 95% prediction interval indicates where 95% of all students' scores in SE would fall, for the appropriate value of ISAD.

3. ANALYSIS OF SOFTWARE ENGINEERING STUDENTS MODULE RESULTS FOR THE THREE COURSES AS ONE MODEL 1998/99

During the academic session 1998/1999 from a student population of 88, a valid sample size of students completing both components (coursework and examination) for the Software Engineering module is 62 (70%). Using the MINITAB statistical package (MINITAB Student Version, 1998), we fitted a statistical model supporting the overall outcome of the Software Engineering module given a prerequisite of Information Systems Analysis and Design, for students enrolled on BSc Computing, BSc Computer Science or BSc Business Information Systems for the academic session 1998/1999. The statistical model is given below (note: SE stands for Software Engineering, ISAD stands for Information Systems Analysis and Design):

$$SE = 33.6 + 0.51 * ISAD$$

The model above suggests that given that a student has the prerequisite of Information Systems and Analysis module before enrolling on the Software Engineering module, the student's expected score for a final module mark for Software Engineering would be in the range (34%, 85%). The model supporting the coursework aspect of the module for the same academic session is thus (SE CWK is the Software Engineering Coursework):

$$SE\ CWK = 51.4 + 0.27 * ISAD$$

The statistical model supporting the coursework aspect of the module during the academic session 1998/1999 suggests that given that a student has the prerequisite of Information Systems and Analysis module before enrolling on the Software Engineering module, the student's expected score for a final coursework mark for Software Engineering would be in the range (51%, 78%).

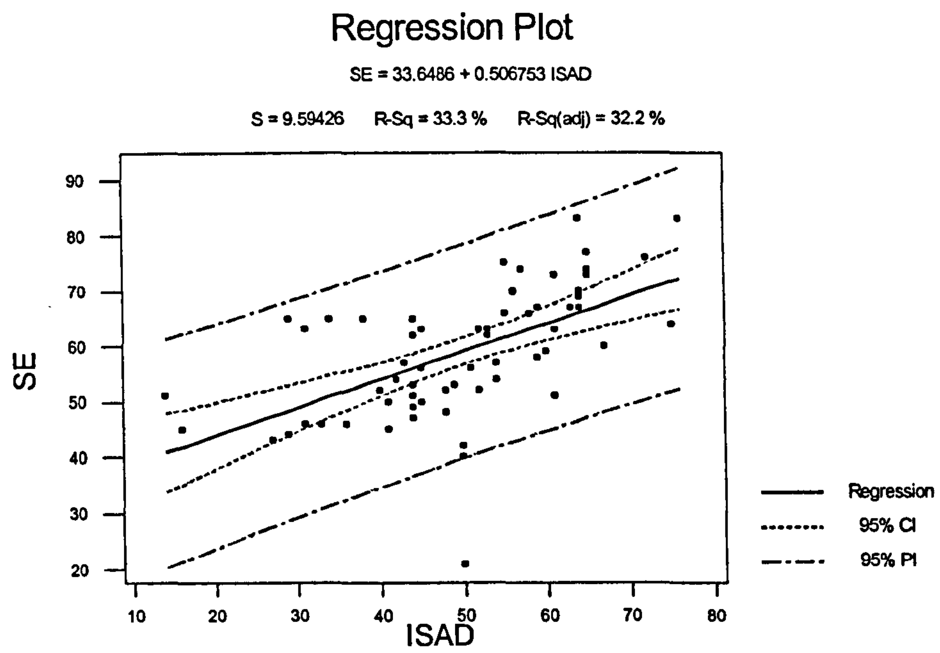


Figure 1. Regression Plot for SE module outcome against ISAD outcome 1998/1999

4. ANALYSIS OF SOFTWARE ENGINEERING STUDENTS COURSEWORK RESULTS BY SOFTWARE LIFE CYCLE PROCESS MODELS 1999/2000

In a previous paper (Oriogun, 2001) we report on a case study designed to show the effect of adopting four different prototyping life cycle models on students' practical coursework in the Software Engineering module during the 1999/2000 academic session. The prototyping methods used are the Win-Win Spiral model, the Spiral model, the Rapid Application Development method (RAD) and the Incremental Development method. The main objective here was to establish an ideal software life cycle process model suitable for adoption within a semester framework for students on BSc (Hons) Computing, BSc (Hons) Computer Science and BSc (Hons) Business Information Systems at the University of North London. The recommendation (Oriogun, 2001) was to address the shortcomings of the Win-Win Spiral model (Boehm 1998) by providing case studies to justify the usefulness and/or effectiveness of the Win-Win Spiral model.

The prerequisites for the module Software Engineering is a pass in an initial programming (first year first semester) module using Visual Basic, and a completion of an Information Systems Analysis and Design module. A detailed analysis of the statistics revealed that the initial programming prerequisite does not have any effect on students' results and it therefore was removed from the statistical models adopted (Oriogun, 2001, p186). The Information Systems Analysis and Design module has a statistically significant effect as a prerequisite for the coursework component of the Software Engineering module. This resulted in the following statistical models (note: SE CWK stands for Software Engineering Coursework, ISAD stands for Information Systems Analysis and Design):

USING THE ONE INTERACTION STATISTICAL MODEL

SE CWK	= 39.94		+ 0.34 * **ISAD**	if Win-Win/Incremental
	= 39.94	-13.86	+ 0.34 ***ISAD**	if RAD
	= 39.94	-11.14	+ 0.34 ***ISAD**	if Spiral

The total number of students adopting the Win-Win Spiral were 26 students. Only 4 students adopted the Incremental Development method, those adopting the original Spiral model totalled 22 and 15 students adopted the RAD method. The Win-Win Spiral and the Incremental Development method were treated together giving a total of 30 students for the first statistical model above. Excluded students include those students transferring from Higher National Diploma (HND) onto the degree programme, students without the two required prerequisites prior to joining the module and those students being reassessed in the module. Therefore, from a population of 98 students who completed the coursework for the module only 67 were used as valid for the study.

The first statistical models above, suggests that the expected results for a student using the Win-Win or Incremental Development method would be in the range (40%, 70%), although any individual student's results would vary about the expectation. The second statistical model suggests that the expected results for a student using the RAD method would be in the range (26%, 60%), although any individual student's results would vary about the expectation, and the third statistical model above, suggests that the expected results for a student using the Spiral model would be in the range (29%, 62%), although again any individual student's results would vary about the expectation.

5. ANALYSIS OF SOFTWARE ENGINEERING STUDENTS MODULE RESULTS FOR THE THREE COURSES AS ONE MODEL 1999/2000

During the academic session 1999/2000 from a student population of 81, a valid sample size of students completing both components (coursework and examination) for the Software Engineering module is 55 (68%). Using the MINITAB statistical package (MINITAB Student Version, 1998), we fitted a statistical model supporting the overall outcome of the Software Engineering module given a prerequisite of Information Systems Analysis and Design, for students enrolled on BSc Computing, BSc Computer Science or BSc Business Information Systems for the academic session 1999/2000. The statistical model is given below (note: SE stands for Software Engineering, ISAD stands for Information Systems Analysis and Design):

$$\mathbf{SE} = 5.5 + 0.61*\mathbf{ISAD}$$

The model above suggests that given that a student has the prerequisite of Information Systems and Analysis module before enrolling on the Software Engineering module, the student's expected score for a final module mark for Software Engineering would be in the range (6%, 67%). The model supporting the coursework aspect of the module for the same academic session is thus (SE CWK is the Software Engineering Coursework):

$$\mathbf{SE\ CWK} = 16.3 + 0.57*\mathbf{ISAD}$$

The statistical model supporting the coursework aspect of the module during the academic session 1999/2000 suggests that given that a student has the prerequisite of Information Systems and Analysis module before enrolling on the Software engineering module, the student's expected score for a final coursework mark for Software Engineering would be in the range (16%, 73%).

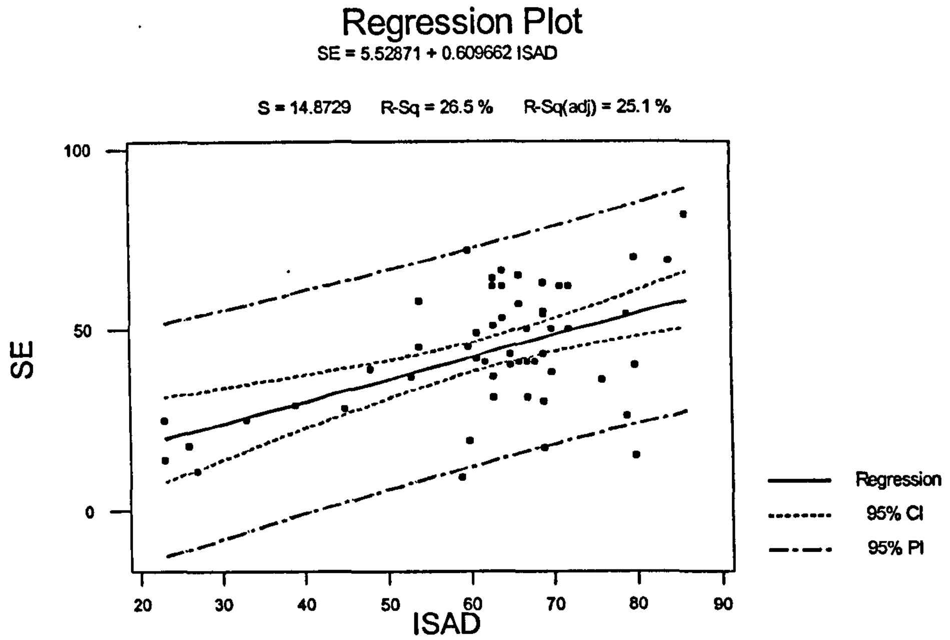


Figure 2. Regression Plot for SE module outcome against ISAD outcome 1999/2000

6. ANALYSIS OF SOFTWARE ENGINEERING STUDENTS MODULE RESULTS FOR THE THREE COURSES AS ONE MODEL 2000/2001

During the academic session 2000/2001 from a student population of 122, a valid sample size of students completing both components (coursework and examination) for the Software Engineering module is 65 (53%). Using the MINITAB statistical package (MINITAB Student Version, 1998), we fitted a statistical model supporting the overall outcome of the Software Engineering module given a prerequisite of Information Systems Analysis and Design, for students enrolled on BSc Computing, BSc Computer Science or BSc Business Information Systems for the academic session 2000/2001. The statistical model is given below (note: SE stands for Software Engineering, ISAD stands for Information Systems Analysis and Design):

SE = 23.4 + **0.45*ISAD**

The model above suggests that given that a student has the prerequisite of Information Systems and Analysis module before enrolling on the Software Engineering module, the student's expected score for a final module mark for Software Engineering would be in the range (23%, 68%). The model supporting the coursework aspect of the module for the same academic session is thus (SE CWK is the Software Engineering Coursework):

SE CWK = 47.6 + **0.22*ISAD**

The statistical model supporting the coursework aspect of the module during the academic session 2000/2001 suggests that given that a student has the prerequisite of Information Systems and Analysis module before enrolling on the Software Engineering module, the student's expected score for a final coursework mark for Software Engineering would be in the range (48%, 70%).

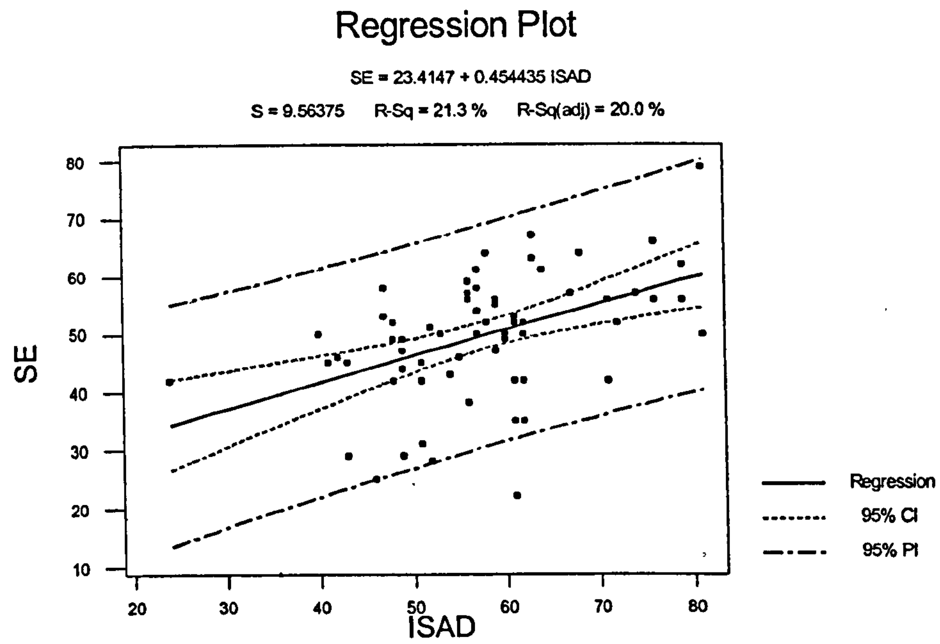


Figure 3. Regression Plot for SE module outcome against ISAD outcome 2000/2001

7. DISCUSSION

During the last three years we have investigated the impact of Information Systems Analysis and Design prerequisite on students' results in a Software Engineering module at the University of North London. The original research question was "would student's performance in an advanced computing module be better enhanced given that they have studied a prior module (at preliminary or advanced) dealing with the basic concepts and groundwork of the advanced computing module?" In order to answer this question, we have looked at a preliminary first year programming module, Visual Basic, and a second year advanced module, Systems Analysis and Design (Oriogun, 2001) as such prerequisites for an advanced starred module (normally studied in the 3rd year at the University of North London, on the undergraduate degree programmes in Computing, Computer Science and Business Information Systems), Software Engineering.

We have chosen the module Software Engineering because three courses (BSc Computing, BSc Computer Science and BSc Business Information Systems) of our undergraduate provision within the School of Informatics and Multimedia Technology at the University of North London, have the software engineering module as compulsory (core) for their pathways. It is therefore an ideal module for our investigation. In our report during the academic session 1998/1999 (Oriogun and Gilchrist, 2000, p401) we concluded that the students' performance on the Software Engineering module could be quite well predicted from the performance on the module Systems Analysis and Design. We suggest further that the teaching of Software Engineering could complement the Information Systems module through its partial use of a particular software life cycle model, and that the skills needed by students to study the two modules are complementary.

We also reported (Oriogun, 2001) on the current research being conducted within the School of informatics and Multimedia Technology, on a possible move towards students' Negotiated Incremental Model -NIM, using case studies from students' coursework results in the module Software Engineering during the 1999/2000 academic session. A detailed analysis of the statistical models supporting the outcome of the coursework element of the module Software Engineering during the same academic session revealed that the preliminary first year programming module does not have any significant statistical effect on students' results as a prerequisite, however, the module Systems Analysis and Design was significant as a prerequisite for the Software Engineering module, thus, confirming our previous results from the academic session 1998/1999.

The reason why the preliminary first year programming did not significantly effect the coursework of Software Engineering could be due to the fact that programming is one of the processes involved in Software Engineering. The life cycle process model adopted in developing a software project and the management of the whole life cycle have an important role to play in terms of the quality of the final software product. It was also revealed that the Win-Win Spiral model and the Incremental Development method are the best process models to adopt for developing the coursework element of the module Software Engineering, although the drawback with the Win-Win Spiral model is that it is documentation intensive. The Negotiated Incremental Model was proposed as a result of trying to find a common ground between the Incremental Development method and the Win-Win Spiral model, in order to facilitate a suitable software process model for students within a semester framework.

The regression lines relating SE CKW to ISAD in 1998/1999 and 2000/2001 are similar. The 1999/2000 regression line shows a different slope, perhaps due to the teaching team being different. A similar pattern is observed for the relationship between SE and ISAD. This study has revealed that on average, students tend to perform a lot better in coursework in comparison to the exam. The average coursework mark during the academic session 1998/1999 is roughly in line with the average coursework mark during the academic session 2000/2001. There was a significant drop in the average mark for coursework and the overall module mark during the academic session 1999/2000.

The reason for the drop in average overall module results and coursework results could be due to the fact that during the academic session 1999/2000, the module leader for the Systems Analysis and Design was off sick half way through the module, and part-time lecturers delivered the rest of the module, which meant that there was no consistency in the teaching and learning of the module. This probably propagated through to the way students performed in the Software Engineering coursework during the same academic session. The module Software Engineering has had the same module leader over the three years of this study. However, the turnover of tutorial assistants for the module has been on the high side, which could have had some effect on the average coursework results over the three years.

8. CONCLUSIONS AND FUTURE WORK

Until the professional computer science student is convinced of the need to use a design method it is unlikely that the idea will be taken up by a lay computer programmer. The result of this study is encouraging and gives initial indication as to the real need for students to have appropriate prior study for advanced computing modules in general. The feedback from software engineering students has been favourable, although they found the coursework aspect of the module to be very demanding given that it had to be completed in ten weeks. They were worried about consistency in marking from one tutorial lecturer to another. However, as there has been a module moderation exercise over the duration of this study, it helped to maintain some consistency in the marking of the group coursework. Students also mentioned that the experience gained on the Software Engineering coursework was of great value to them during their final year projects.

There seems to be a proliferation of names for courses within universities, all of which teach the same basic topic in software engineering. There needs to be research into existing software engineering courses at universities within the UK to uncover this nucleus. This research material will form the basis of our future work into finding appropriate prior study for software engineering courses, starting from year one software engineering modules through to final year software engineering modules.

REFERENCE

Boehm B W. (1998). *A Spiral Model of Software Development and Enhancement*, Computer, vol 21,no 5, May 1998, pp61-72.

Capability Curriculum (2001).
http://www2.unl.ac.uk/~head/capabil.htm

Kelly J. (2000). *Information Systems Design: enabling students to be more creative*, Proceedings, UKAIS 2000, UWIC, 26th -28th April 2000, McGraw Hill, ISBN 0077097556, pp211-218.

MINITAB Student Version. (1998). *Release 12 for Windows*, Minitan Inc.

Oriogun P K. (2000). *The Development of a Computing Module incorporating Capability Curriculum*, Education Media International, 37(4) 2000, ISSN 0952-3987.

Oriogun P K and Gilchrist R (2000). *The Effect of Studying an Information Systems Module on students' Results in a Software engineering Module*, 5th UKAIS Conference, UWIC, 26th -28th April 2000, McGraw Hill, ISBN 0077097556, pp 394-401.

Oriogun P K. (2001). *Towards Students' Negotiated incremental Model -NIM: A Case study on Students' In-Course-Assessment in a Software Engineering Module*, 6th UKAIS Conference, University of Portsmouth, 18th - 20th April 2000, pp175-189.

Software Engineering (2001).
http://www.unl.ac.uk/simt/im283/

ITALICS is now in the process of being re-launched and will be fully functional by January 2004. Information regarding ITALICS, including submission details, can be obtained by contacting Gillian Livesey Tel: +44(0) 2890 366109

ITALICS

Innovations in Teaching And Learning in Information and Computer Sciences

The e-Journal of the LTSN-ICS

RTF VERSION

Towards Understanding Software Requirements Capture: Experiences of Professional Students using the NIA to Support the Win-Win Spiral Model

Peter Oriogun
Learning Technology Research Institute,
University of North London, UK

Email:p.oriogun@unl.ac.uk

Abstract

This paper presents a critical appraisal of the Win-Win Spiral model as a possible process model for developing small-scale software from inception to implementation within a semester framework at higher education institutions. It also presents the Negotiated Incremental Architecture (NIA) and its decision rationale, developed at the University of North London as being a possible architecture that supports the Win-Win Spiral model with respect to developing software within higher education institutions over a semester time frame. A group of mature students undertook a coursework, which involved implementing a reasonably straightforward web-based application using the Win-Win Spiral model [1,2]. The task was to produce an on-line golf league system where players could input match results and view their league rankings on the web. The coursework was a requirement for the module 'Software Engineering' on the postgraduate masters programme at the University of North London, during semester A of 2001/2002 academic session. Having used the Win-Win Spiral model, we consider the model to be too heavy on documentation and will propose to support the on-going research at the University of North London, to develop a collaborative model, which is an adaptive hybrid of the Win-Win Spiral and the Incremental Development. This proposed Negotiated Incremental Model [3,4,5] will prove to be attainable within a semester framework as well as realistic to implement in industry.

1. Introduction

The Win-Win Spiral model also known as the Theory-W Based Spiral model has its theoretical basis in management theory [18]. The theory argues that a project will be successful if and only if the critical stakeholders (users, customers, developers and maintainers) are all "winners", thus the term Win-Win. If the requirements of any one of these stakeholders are omitted, then a win-Lose situation pertains -for example, where a customer's requirements are met but not those of the users. Such Win-Lose scenario is, in reality, a Lose-Lose situation because a project can only be successful if all major parties achieve their essential objectives. A typical one could be a situation whereby a piece of software meets the customer's cost requirements but does not actually work according to the user's requirements. The "winner" also looses since s/he is going to have to either put up with a less than optimal performance or be faced with further expense of developing the software [19].

The golf league project was based on a coursework allocated to MSc Computing students in a Software Engineering module at the University of North London (UNL). It has been observed [4] that group of students who implemented the Win-Win Spiral model have achieved better final module results than those students using the Rapid Application Development [6] or Incremental Development [6] models. The group also felt the Win-Win Spiral model was an innovative and progressive approach to software engineering activities. The time allocated for the project was one semester, which was a rather brief timeframe to develop the prescribed software from analysis through to implementation, considering that the model requires the production of three major documents, namely, the Life Cycle Objective (LCO), the Life Cycle Architecture (LCA) and the Initial Operational Capability (IOC) [2,7,8].
In this paper we discuss the experience of a group of mature none computing professionals (including a medical doctor, a business analyst, an experienced educator, a building construction specialist, a social scientist, and an experienced business administrator) undertaking postgraduate studies in Computing at the University of North London during the academic session 2001/2002.

2. Choice of Software Prototyping Method

This section describes the features of the available software process models to be adopted for the development and implementation of the coursework project [5,6]. The models are as follow:
- Incremental Development
- Waterfall
- Spiral
- Win-Win Spiral

2.1 Incremental Development Method

Each increment of the software produces a working (releasable) version of the software. It is particularly practical for the development of applications designed for commercial distribution such as word processing systems, personnel and business accounting systems. While market research, conducted in an attempt to ascertain what features the potential users may want or need, forms a good basis for design, by the time testing is conducted just prior to release, ideas for added features, design improvements and additional requirements are likely to emerge. In a sense, each working release functions as a prototype for future versions. The Incremental Development and Win-Win Spiral models for software development bear many similar attributes.
- Both are iterative approaches, and each allows the capture of emerging requirements, the incremental model through feedback from the users and the Win-Win spiral as it moves through each cycle.
- Both allow for design improvements. Design improvements can be accommodated as improved hardware, software programming techniques and financial resources become available.
- Both provide for testing of commercial viability of the product with minimal financial risk. A concept product with basic features can be released and the uptake measured. Consumer interest will influence the amount of resources to commit to future versions.

2.2 Waterfall Model

The Waterfall model is based on the belief that there are different stages to the development of software. Outputs from one stage flow into the next. There are usually five stages in the development of software [16].
- Requirements phase
- Design phase
- Implementation
- System Testing
- Operation and Maintenance

Documentation is intrinsic to the Waterfall model. The output of each stage is documented and verified. The software process is not always linear. When errors in earlier stages are discovered the development is sent back to that stage. It may be acceptable to go back one stage, but it can be disastrous for a project if it has to go back several stages, as requirements are specified early in the development cycle and it becomes difficult to change them at a later stage. Because of this feature, the Waterfall model is best suited to projects where the users have a good understanding of what their requirements are. The waterfall model is a better choice in comparison to other software life cycle process models when the development team has done similar projects in the past as they will have a better idea of the final product and can convey that to the users.

2.3 Spiral Model

The Spiral model provides the potential for rapid development of incremental versions of the software being developed. Software is developed in a series of incremental releases. The early iterations of the incremental release may be a paper model or prototype. Later iterations produce increasingly complex, more complete versions of the software system. The Spiral model is divided into a number of framework activities, or task regions. Typical models contain between three and six task regions [1,6]:
- Customer Communication
- Planning
- Risk Analysis
- Engineering
- Construction and Release
- Customer Evaluation

As the project progresses the development team moves around the spiral. The first circuit may result in the development of a product specification. Subsequent passes around the spiral might be used to develop a prototype and then progressively more sophisticated versions of the software. Each parse through the planning region results in adjustments to the project plan. The key to the original Spiral model [1] is that the software evolves as the process progresses; the developer and customer better understand and react to risks at each evolutionary level. Prototyping is used as a risk reduction mechanism and enables the developer to apply the prototyping approach at any stage in the evolution of the product.
The drawback of the Spiral model is that it demands considerable risk management expertise and relies on this expertise for success. If a major risk is not uncovered and managed, problems will occur. The Spiral model sets out a framework for customer communication. In an ideal situation the customer tells the developer what they want and the developer goes away and develops the software. However, real life is not as straightforward as this, and this is where the Win-Win spiral model refines the spiral model.

2.4 The Win-Win Spiral Model

The Win-Win Spiral model [2,7,8] is a recent example of a software process model. It is a risk-driven process, which uses a cyclical approach for incrementally growing a system's degree of definition and implementation. Its success criterion is to achieve stakeholder concurrency in developing a system. It has three main distinguishing features. First, the Win-Win spiral model uses a cyclic approach for incrementally growing a system's degree of definition and implementation. Secondly, it is a model of a process based on Theory-W [9,17], which is a management theory and approach based on making winners of all of the system's key stakeholders as a necessary and sufficient condition for project success. Finally, it incorporates a set of anchor point milestones for ensuring stakeholder commitment to feasible and mutually satisfactory system solutions [8]. It embraces the Unified Process through the use of the anchor points, adopting the UML notation for specifying and modeling the software to be implemented.

3. The Anchor Points for the Golf Project

The Spiral Model has been extensively elaborated, and successfully applied in numerous projects. However, some common difficulties such as determining where the elaborated objectives, constraints, and alternatives come from, have led to some further extensions to the model. Hence, the Win-Win Spiral model resolves this by adding the Win-Win process activities, detailed below, to the front of each spiral cycle;

- Identify the system or subsystem's key stakeholders.
- Identify the stakeholders' win conditions for the system or subsystem.
- Negotiate Win-Win reconciliation of the stakeholders' win conditions.

The first activity of the group was to identify the three key stakeholders:

- Club and league Officials
- Golfers
- Developers

The group then identified Win conditions for the project. Forty-three win conditions were negotiated. Most of these conditions specified functional and non-functional requirements of the proposed system. Some of the players win conditions were that the system should be on-line, allow browsing of results and ranking, and use a points system for comparative ranking. Therefore when the group came to compile the Operational Concept Definition (OCD) [10], they found most of this was specified in the Win conditions.

The Win-Win model has also added a set of life cycle anchor points to the Spiral Model namely the Life Cycle Objective (LCO), Life Cycle Architecture (LCA) and Initial Operational Capability (IOC) to synchronize the spiral cycle with shareholder commitment and critical management decision points. [11].

A key feature of the LCO milestone is the Feasibility Rationale. This demonstrates a viable business case for the proposed system. Feasibility Rationale is essential as costing and estimation of time required for projects determines weather a project will go ahead. As the project was a learning exercise the relevance of this section was undermined. However, costing was estimated using CoCoMo 81 [7]. Estimation of time required using COCOMO was irrelevant as a hand-in date was specified for the coursework. Also the time frame calculated by CoCoMo assumes a 'real industry project' based on complete and thorough processing of all aspect of software engineering.

The whole question of whether the project was feasible was not relevant - the group had to make an attempt at the project in order to complete the module. The LCA is largely an expansion and elaboration of LCO's, and specifies a feasible architecture. The feature that distinguishes the LCA from the LCO is the need to have major risks resolved. Again the group did identify some key risks, but the overall impression of the group was that risk analysis was not as relevant to student coursework as it would be in industry.

The group approached the LCO and LCA anchor points by dividing them into milestone elements [10]:

- Operational Concept Description (OCD)

- Requirements Definition (SSRD)
- Architecture description
- Life cycle plan
- Feasibility rationale

The third anchor point, called the Initial Operational Capabilities (IOC), defines the operational capability, including the software environment needed for the first product release, operational hardware and site environment, and customer manuals and training.
The IOC anchor point involves:
- Software preparation (operational and support software and necessary licenses for COTS)
- Site preparation
- User, operator and maintainer preparation

The IOC is again largely irrelevant to our golf project. No reused operational software was utilized, site preparation is not relevant as the University provided support software and hardware required. End user training and familiarization with the product are not important, as there is no end user for a teaching module project, except for the module leader -who may have other use for the software for future research purposes within the School of Informatics and Multimedia Technology.

While this amount of documentation maybe suitable for large, long-term projects, the group found the documentation to be excessive for small project over a semester. The group's overall impression was that the Win-Win Spiral model has many checks and balances and is a very thorough and rigorous software process model. However, a great deal of documentation was produced for a small-scale project and much of the documentation was repetitive. On the positive side, the group learnt that the processes involved in developing a software product is just as important as the end product itself.

4. Using The Win-Win Spiral Model for Developing a Software Engineering Project Within One Semester

The group was given 10 weeks to research and implement a golf league ranking system using the Win-Win Spiral process model whilst still continuing the other modules of the MSc programme at the University. This proved to be a very difficult task. Given the time constraint and the fact that the coursework was set using a Problem-Based Learning (PBL) approach [14,15]. The group found it necessary to adapt the Win-Win Spiral architecture in such a manner that the time of each cycle could be reduced.

The Win-Win Spiral Model uses a negotiation router (conferencing support tool such as QARCC - Quality Attribute & Conflict Consultant) as a point of contact between Stakeholders [8]. In Industry it may be convenient and a time saving measure for individuals to communicate through a conferencing support tools. For instance, it maybe impossible for stakeholders to travel to every meeting, or perhaps stakeholders who do not have a lot to contribute or take from the meeting would not wish to attend meetings etc. However, in a situation where time is critical and an incremental negotiation model involving the stakeholders is desired, the complete co-operation, ownership and dedication of all stakeholders is essential. This was effectively the situation with the set coursework, with students simulating the roles of stakeholders.

5. The Negotiated Incremental Architecture (NIA)

As detailed earlier the Win-Win Spiral model is 'document heavy' and managing this process was critical to the success of the project. In order to meet the time constraint the group adapted the Win-Win Spiral architecture by having 'Face to Face' meeting and using a publicly available conferencing facility, Yahoo-Group [12] to post documents for the group for individual reviews. The Negotiated Incremental Architecture (NIA) was adopted in order to simulate the Win-Win architecture. See Figure 1 for the NIA as proposed in [20] at the University of North London.

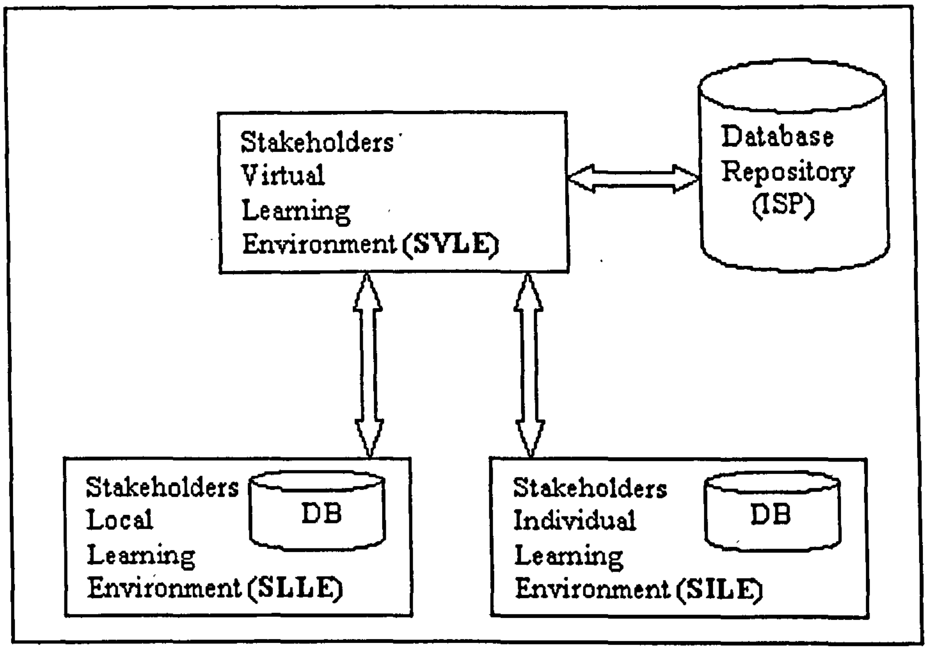


Figure 1: The Negotiated Incremental Architecture -NIA

5.1 Stakeholders Virtual Learning Environment -SVLE

Students were asked to join an Internet Service Provider (ISP) such as Yahoo Groups in order to facilitate the reconciliation phase of the Win-Win. This is an attempt to resolve conflicts between win conditions. As a result of setting up a negotiation facility away from the university, students could continue their negotiations when the university server is down. The database repository linked to the SVLE represent the fact that other users may have their information stored with the same ISP. The module tutor is an essential stakeholder, as it is necessary for the student stakeholders to seek clarification in parts throughout the negotiation phase of the software development process. The student stakeholders are encouraged throughout the negotiation phase to play active roles, and rotate the management of the group on a two weekly basis in order for every member of the group to have ownership of the developing software.

5.2 Stakeholders Local Learning Environment -SLLE

The Stakeholder Local learning environment is where the students come together as a group at the university to have a face-to-face meeting, about the requirements, as well as recording it online for future reference. At this stage, there may be one member of the group appointed to record the proceedings of the meeting and post it at a later date or immediately onto the SVLE database. The module tutor (who is also one of the stakeholders) could be part of the SLLE negotiation phase otherwise the tutor could access the student stakeholders as within his/her own SILE via the SVLE set up by the group.

5.3 Stakeholders Individual Learning Environment -SILE

The Stakeholder Individual Learning Environment (SILE) depicts the negotiation carried out by each stakeholder away from the university, either by using their own home computer(s) or using other computer(s) to access the SVLE in order to obtain updated information as to the negotiation activities thus far within the group of stakeholders. The SILE is primarily via a computer, however, it is possible that students could access the SVLE via a mobile telephone as well as other telephone lines.

5.4 Negotiated Incremental Architecture Schematic Logic

If a win condition is non-controversial, there is no conflict (C), the win condition covered by Agreed Negotiated Win Condition (E). Relationship between the win conditions are established, leading to issues relating to the Controversial Win Conditions (D) being identified which raises the conflicts between win conditions and their associated risks and uncertainties. Options are considered (On) which suggest strategies for resolving issues, which lead to Agreed Negotiated Win Condition (E) that satisfies stakeholders win conditions and also defines the systems objectives. Any unresolved win conditions is then considered at the next cycle of negotiation (B), otherwise it has been agreed at the First Cycle of Negotiation (A). Figure 2 shows the decision rationale for the Negotiated Incremental Architecture.

```
IF FIRST CYCLE NEGOTIATION (A)
        IF    NON CONTROVERSIAL WIN CONDITIONS (C)
              AGREE NEGOTIATED WIN CONDITIONS (E)
        ELSE
}             CONTROVERSIAL WIN CONDITIONS
}             WAIT UNTIL NEXT CYCLE NEGOTIATIONS
          WHILE OPTIONS AVAILABLE  (On)
                CONTINUE NEXT CYCLE NEGOTIATIONS (B)
                REVISE OPTIONS AVAILABLE (On)
             LOOP UNTIL
                NON CONTROVERSIAL OPTIONS FOUND (C)
                AGREE NEGOTIATED WIN CONDITIONS (E)
             END LOOP
          ENDWHILE
        ENDIF
ENDIF
```

Figure 2: Negotiated Incremental Architecture Schematic Logic

6. Discussion

This section will discuss and summarise the group's findings and suggest ways in which the group would have approached the coursework differently with the benefit of hind site.
Identifying stakeholders and negotiation between those stakeholders plays a critical role in the Win-Win process. The group found that this provided many benefits. By allowing stakeholders to present their win conditions and negotiate with other stakeholders regarding their win conditions results in the key people buying into the product. Accordingly, this reduces the risk of product rejection at later stages. The iterative negotiation process also provides an opportunity for the win-conditions to be revisited to reflect changes in priorities or market conditions.
However, these advantages need to be balanced with time and the cost of late changes. In the group's opinion the process of revisiting win conditions could only be beneficial if managed effectively. By this we mean that allowing a mechanism for re-visiting win-conditions that affect the functionality of the system could result in time for implementation being extended or the start time to be forced back. If changes are made during implementation costs can increase. Therefore, it is important that only important changes are allowed during the negotiation iterations, and whimsical desires quashed.
There are obvious dilemmas involved with developers not wishing to upset the client, but the skill of the negotiator comes into play by tactfully managing discussions and clearly identifying the knock on effects of any changes, e.g. increased functionality versus time to market. Completing the coursework within one semester meant that time was critical. The Win-Win Spiral Architecture supports the use of a negotiating router (conferencing support tool). Although the benefits of using such a conferencing tool in a 'real world' situation was recognised, it would have proved unworkable with the coursework at hand. Instead the group adopted a different architecture, favouring 'face-to-face' meetings by all Stakeholders (students) to discuss, negotiate and agree the win conditions as well as the way forward. The group used Yahoo-group to place documentation for all to view.

To a large extent the functional win conditions focused around the groups programming capabilities. In a real world situation developers would have been expected to have competent programming skills and as to this extent negotiations did not reflect a realistic situation. The group was of the opinion that a Win-Win situation, as the name of the model suggests, is rarely achieved. In many instances the process of negotiation helped to ensure that a Win-Loose or Loose-Loose situation did not occur. Compromise-Compromise would be a more apt, although not as catchy a name for the process.

7. Conclusions / Future Work

Having enrolled on a conversion Masters degree in computing the group had little to no experience of software engineering, let alone the Win-Win Spiral model. Researching the Win-Win model and realising the three main anchor points of the model, namely the LCO, LCA and IOC, proved extremely time consuming. The group viewed the LCA and IOC as an expansion and detailed discussion of the LCO's and served as a checking measure, as well as the basis of selection for the preferred approach. This tended to produce a lot of redundant and duplicated documentation and the group decided to refer to the LCO's where appropriate.

As a consequence of the group's inexperience with the discipline of Software Engineering, in particular the Win-Win Spiral model, the time expended on the LCO's was drastically longer than that allocated in the plan. This time overrun had a strong impact on the implementation time, resulting in reduced quality and reduction in the extent of the implementation. Although the advantages of the LCA and IOC phases were recognised, the group was of the opinion that these phases could be dropped from the process by selecting a feasible architecture on the basis of a superficial review of the LCO's, this would increase time for implementation.

Many other aspects of the Win-Win, such as feasibility study, risk analysis, CoCoMo, COTS etc. were given minimal attention, as they were not considered relevant in the context of an academic coursework. In reality these are extremely important commercial considerations and form a critical element of managerial decision- making. Had a detailed exploration of these aspects been deemed necessary within the allocated timeframe the group feel the quality of the remainder of the coursework would have suffered. A high quality document and fully implemented system could have been completed using the Win-Win Spiral model if the group were experienced with the model prior to starting the coursework as well as having knowledge of web design.
An important consideration when selecting a software process model is the quality required against the products time to market. We support the on-going research at the University of North London, to develop a collaborative model, which is an adaptive hybrid of the Win-Win Spiral and the Incremental Development. This proposed Negotiated Incremental Model using the NIA and its decision rationale should prove to be attainable within a semester framework as well as realistic to implement in industry.

Acknowledgements

I would like to acknowledge the following students who made invaluable contributions to this article:

N S Tilooh, J P Hansen, P Assawaniwest, K N Ranasinghe, A Russell and R Anandasangaree.
References

1. Boehm B W, (1988). A Spiral Model of Software Development and Enhancement, Computer, May, 1988, pp 61-72.

2. Boehm B W, (2000). Spiral Development: Experience, Principles and refinements, USC Spiral Experience Workshop, 9th February 2000.

3. Oriogun P. K , (2000a). The Development of a Computing Module Incorporating Capability Curriculum, Education Media International Journal, 37(4)2000, ISSN 09523987.

4. Oriogun and Mikusauskas, (2001). Win-Win Decision Support Framework: A Case Study on Students' In-Course-Assessment in a Software Engineering Module, Proceedings of the ICEIS 2001 Conference, ISBN 972-98050-2-4, vol 1 pp370-376.

5. Roberts, Zaouini and Swann, (2001). An On-line Simulated Golf League Multimedia Environment using the Win-Win Spiral Model: A Case Study for a Postgraduate Software Engineering Module, Proceedings of SQM 2001, ISBN 1-902505-40-9, pp309-323.

6. Pressman R S, (2000). Software Engineering - A Practitioner's Approach, 5th Edition, European Adaptation, McGraw Hill, pp 26-40.

7. Oriogun P. K , (1999). A Survey of Boehm's work on the Spiral Model and COCOMO II -Towards Software Development Process Quality

Improvement, Software Quality Journal, 8, 53-62 (1999), Kluwer Academic publishers, USA.

8. Boehm, BW. Using the WinWin Spiral Model: A Case Study, Computer, vol.31, no.7, July 1998, pp 33-44].

9. Boehm, Bose and Horowitz et al. (1994). Software Requirements as Negotiation and Regeneration Aids: A Theory-W Based Approach, Proceedings, 17th ICSE Conference, pp1-12.
10. Horowitz et al. (1997). WinWin Reference Manual: A System for Collaboration and Negotiation of Requirements, CSE, University of Southern California, 1997.

11. Boehm, Egyed, Kwan and Madachy (1997). Developing Multimedia Applications with the WinWin Spiral Model. [On-line] http://sunset.usc.edu/TechRpts/Papers/usccse97-504/usccse97-504.html

12. Yahoo Home Page [on-line] available http://www.yahoogroups.com

13. Problem Based Learning Institute.(2001). [Online] available:

14. Oriogun and Georgiadou (2000). Towards making Learning Agents Think of Their roles in the Learning Process through a Problem-Based Learning Grid, Proceedings of the, 2nd Asia pacific conference on Problem Based Learning, [On-Line] http://www.tp.edu.sg/pblconference/full/OriogunGeorgiadouPaper.pdf

15. Oriogun, Ferguson and Ludmann et al. (2001). Using the Problem-Based Learning Grid in the Management of Software Engineering Module: Towards an Effective Learning Intervention for final Year Students, In, The Power of Problem Based Learning -Experience Empowerment Evidence, (Eds) Conway and Kandlbinder, PROBLAC, Australia Problem Based Learning Network.

16. Royce W. W (1970). Managing the development of large software systems: concepts and techniques, Proceedings, WESCON, August 1970.

17. Oriogun P. K (2000b). A Review of Theory-W Based Spiral Process Model and its associated Tools for Supporting Conflict Identification and Negotiation in Software Development, Proceedings, SQM 2000, BCS Publication, ISBN 1-902505-5, pp53-64.

18. Boehm and Ross (1989). Theory W Software Project Management: Principles and Example, IEEE Trans. On Software Engr., July 1989, pp. 902-916.

19. In, H (1998). Conflict Identification and Resolution for Software Quality Attribute Requirements, PhD Thesis, University of Southern California,1998.

20. Oriogun et al. (2002). Towards software Negotiation Process Improvement:

Experiences of Professional Students using the Win-Win Spiral Model, Proceedings, INSPIRE 2002, BCS Publication, ISBN 1 902505 46 8, pp91-106.

CONTENT ANALYSIS OF ONLINE INTERRATER RELIABILITY USING THE TRANSCRIPT RELIABILITY CLEANING PERCENTAGE (TRCP): A SOFTWARE ENGINEERING CASE STUDY

Peter. K. Oriogun
London Metropolitan University, London North Campus,
166-220 Holloway Road, London N7 8DB
Department of Computing, Communications Technology and Mathematics, London, England
Email: p.oriogun@londonmet.ac.uk

Keywords: Content Analysis, Computer Mediated Communication, Cohen's kappa, Vicarious Learning, Win-Win Spiral Model, Virtual Learning Environment, Cooperative Systems for Learning

Abstract: In this paper the author presents a case study of online discourse by message unit using quantitative content analysis, with particular emphasis on the author's proposed interrater agreement percentage which is referred to in this paper as the Transcript Reliability Cleaning Percentage (TRCP). It is argued in this paper that our proposed Transcript Reliability Cleaning Percentage (TRCP) will better enhance interater reliability in the case whereby categorisation of online messages is decided upon after the collation of all the required participants' messages for examination. We examined the ratings of participants' online messages, at the end of the semester, in terms of level of engagement within a negotiation forum in line with the author's Negotiated Incremental Architecture, Oriogun (2002). Categorisation of the messages was determined at the end of the semester in consultation with the second rater of the transcripts, after perusing though the first 30 messages. The variables that the author investigated are, *participation*, and *interaction*. The paper is divided into six sections, that will introduce the rationale for the study, a brief introduction to the Negotiated Incremental Architecture, followed by the study itself, we then define what we means by Transcripts Reliability Cleaning Percentage (TRCP) of online discourse using message unit, followed by the interpretation of individual participant's result and finally the author will conclude with the impact of our results on the design of cooperative systems for learning; project management systems for online negotiation of requirements; software engineering students in higher education negotiating software systems requirements, and a recommendation of a follow-on paper, using an approach we are currently working on at the London Metropolitan University called the SQUAD approach to on-line discourse. The SQUAD approach is a semi-structured categorisation of online messages.

1 INTRODUCTION

The impetus for this paper stems from reading the article by Rourke, Anderson, Garrison & Archer (2000), titled, Methodical Issues in the Content Analysis of Computer Conferencing Transcripts. The author was faced with a similar scenario as the one presented in the article relating to the "Professor Jones's of this world". It has certainly released the "educational treasures that the authors of the article believed are locked in the transcripts that document learning in the online environment".

One of the courses that the author teaches at London Metropolitan University is titled Software Engineering for Computer Science. This course is a compulsory course for three named undergraduate degree pathways, namely, BSc Computing, BSc Computer Science and BSc Business Information Systems. The course typically attracts between 100-130 students (including students retaking the course because they have failed in the previous year). In the academic year 2001/2002, there were a total of 95 students completing the course. The 95 students were split randomly into 15 tutorial groups and each group on average is composed of 6 members. Each group was assigned a designated tutorial assistant. There were three assessments for the course, comprising of, a group coursework (40%), individual coursework (10%) and final examination (50%). The group coursework also had an individual element attracting a maximum of 13% of the marks. This individual element is the online negotiation of software requirements. Each group member had to contribute to the online negotiation throughout the period of the course, which was one semester, a total of 12 weeks. As a way of getting students to take the online assessment seriously, each group member had to attach up to a maximum of 10 of their posting to the conference forum, as part of the final group report. The marking scheme for the group coursework did not involve the use of content analysis, instead, they scored most of the marks for the online assessment on the basis that they submitted the required 10 posting.

The author was interested in measuring the quality of his students' online negotiation of software requirements. The author was also interested in finding out the reasons why there has been very little research on interrater reliability with respect to content analysis of online discourse. The case study presented in this paper is a way of answering all these questions. The author instructed all his students on the course in the academic session 2001/2002 to make use of an external Internet Service Provider for the online part of the course. Majority of the students had used Yahoogroups (2002). The author extracted all the postings from one particular group into a word document, and proceeded to investigate two variables, namely, each student's *participation* and *interaction* using message as unit of analysis.

2 ONLINE DISCOURSE USING THE NEGOTIATED INCREMENTAL ARCHITECTURE

In order to meet the deadline of ten weeks for the coursework, the group adopted the negotiated Incremental Architecture -NIA, by having 'Face to Face' meeting and using a publicly available Internet Service Provider, Yahoogroups (2002) to post documents for the group for individual reviews. The Negotiated Incremental Architecture was adopted in order to simulate the Win-Win architecture (Boehm 1988; Oriogun 1999), as the process model that was allowed for modelling and developing the coursework was Win-Win Spiral model (Boehm 1988; Oriogun 1999). See Figure 1 for the NIA as proposed by the author at the London Metropolitan University.

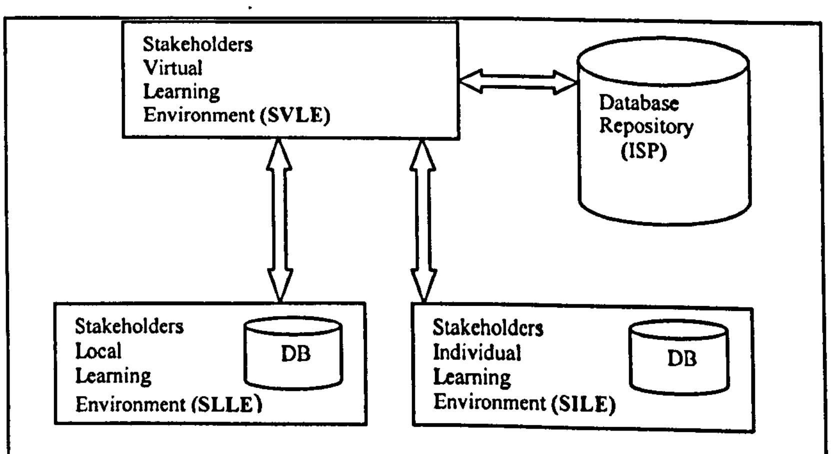


Figure 1: Negotiated Incremental Architecture -NIA

2.1 Stakeholders Virtual Learning Environment -SVLE

Students were asked to join an Internet Service Provider (ISP) such as Yahoo Groups in order to facilitate the reconciliation phase of the Win-Win. This is an attempt to resolve conflicts between win conditions. As a result of setting up a negotiation facility away from the university, students could continue their negotiations when the university server is down. The database repository linked to the SVLE represent the fact that other users may have their information stored with the same ISP. The module tutor is an essential stakeholder, as it is necessary for the student stakeholders to seek clarification in parts throughout the negotiation phase of the software development process. The student stakeholders are encouraged throughout the negotiation phase to play active roles, and rotate the management of the group on a two weekly basis in order for every member of the group to have ownership of the developing software.

2.2 Stakeholders Local Learning Environment -SLLE

The Stakeholder Local learning environment is where the students come together as a group at the university to have a face-to-face meeting, about the requirements, as well as recording it online for future reference. At this stage, there may be one member of the group appointed to record the proceedings of the meeting and post it at a later date or immediately onto the SVLE database. The module tutor (who is also one of the stakeholders) could be part of the SLLE negotiation phase otherwise the tutor could access the student stakeholders as within his/her own SILE via the SVLE set up by the group.

2.3 Stakeholders Individual Learning Environment -SILE

The Stakeholder Individual Learning Environment (SILE) depicts the negotiation carried out by each stakeholder away from the university, either by using their own home computer(s) or using other computer(s) to access the SVLE in order to obtain updated information as to the negotiation activities thus far within the group of stakeholders. The SILE is primarily via a computer, however, it is possible that students could access the SVLE via a mobile telephone as well as other telephone lines.

2.4 Negotiated Incremental Architecture Schematic Logic

If a win condition is non-controversial, there is no conflict (C), the win condition covered by Agreed Negotiated Win Condition (E). Relationship between the win conditions are established, leading to issues relating to the Controversial Win Conditions (D) being identified which raises the conflicts between win conditions and their associated risks and uncertainties. Options are considered (On) which suggest strategies for resolving issues, which lead to Agreed Negotiated Win Condition (E) that satisfies stakeholders win conditions and also defines the systems objectives. Any unresolved win conditions is then considered at the next cycle of negotiation (B), otherwise it has been agreed at the First Cycle of Negotiation (A). Figure 2 shows the decision rationale for the Negotiated Incremental Architecture.

```
IF FIRST CYCLE NEGOTIATION (A)
        IF          NON CONTROVERSIAL WIN CONDITIONS (C)
                    AGREE NEGOTIATED WIN CONDITIONS (E)
        ELSE
}                   CONTROVERSIAL WIN CONDITIONS
}                   WAIT UNTIL NEXT CYCLE NEGOTIATIONS
          WHILE OPTIONS AVAILABLE (On)
                      CONTINUE NEXT CYCLE NEGOTIATIONS (B)
                      REVISE OPTIONS AVAILABLE (On)
            LOOP UNTIL
                      NON CONTROVERSIAL OPTIONS FOUND (C)
                      AGREE NEGOTIATED WIN CONDITIONS (E)
            END LOOP
          ENDWHILE
        ENDIF
ENDIF
```

Figure 2: Negotiated Incremental Architecture Schematic Logic

3 RESEARCH QUESTIONS

Before engaging with this research, the author was primarily interested in answering three questions at the end of the research, with the hope that it will lead to future exciting research in the area of content analysis of computer conference transcripts. The questions are as follow:

(i) Is it possible to measure the quality of online discourse?

(ii) Is there an alternative or better reliability measure than the current percentage agreement and Cohen's kappa value for computer conference transcripts?

(iii) Why has there been very little research conducted on interrater reliability in relation to content analysis of computer conference transcripts in the past two decades?

4 THE STUDY

The case study presented in this paper is based on the online negotiation of software requirements. Seven students and the module lecturer participated in the study. The students will be referred to as Students 1, Student 2, Student 3, Student 4, Student 5, Student 6 and Student 7 respectively. The Lecturer will be referred to as the tutor in this paper. There were a total of 141 messages posted over a period of 95 days, including weekends. In order to investigate the contribution of each student in terms of *participation* and *interaction* during the online negotiation, it was necessary to extract the messages from Yahoogroups (2002) onto a word document before processing of the messages. The problem with using such ISP is that there are advertisements embedded within each message, which meant that it took about 6½ hours to extract all the messages.

In order to start the analysis, the author carefully constructed categories of coding decisions, and each coding decision needed to be rated in some manner. The coding decision was based on the type of messages that the author deemed to involve *participation* and *interaction*. A total of 9 coding decision were generated, each coding decision also has a rating attached to it. The ratings are from 0 to 8, where 0 means very little or no *participation* and *interaction*, and 8 indicate high *participation* and *interaction* in the author's judgement Table 1 below shows the coding decision based on message rating.
The coding of the transcripts took 2 hours 40 minutes, which translates to around 1.135 minutes per message. The coding of the initial TRCP took the author 1hour 10 minutes (see Table 2 below). The coding of the final TRCP together with the discussion between the two raters took 1 hour 30 minutes. It would have taking considerably longer if the variables being investigated had been greater than the two focussed on for this study. Furthermore, Table 1 would have been constructed differently, with possibly more categories and sub categories involved. In the case whereby cohen's kappa Cohen (1960) and coefficient of reliability Holsti (1969) are to be calculated, a second coder would be required. These are some of the reasons why researcher who have used quantitative content analysis technique have described it as been difficult, frustrating and time-consuming. The technique has been described as "a research technique for the objective, systematic, quantitative description of the manifest content of communication" (Berelson, 1952, p. 519).

Table 1: Coding Decisions Based on Message Ratings

Coding Decision (Category)	Unit of Analysis (Message)	Rating (R)
1	No engagement with the group	0
2	Agreeing with others without reasons	1
3	Agreeing with others with reasons	2
4	Referring the group to relevant websites	3
5	Resolving conflicts within the group	4
6	Taking a lead role in discussion	5
7	Offering to deliver artefact(s)	6
8	Offering alternative solutions to group problems	7
9	Active engagement with the group	8

Rating of each message is subject to influence by coders, this is the reason why Berelson (1952) stipulates that content analysis is an objective technique.

4.1 Defining Participation and Interaction for this Study

The two variables we investigated in this paper are *participation* and *interaction*. Our definition of *participation* is in line with the definition proposed by Henri (1992). She is one of the early authors to develop a tool to be used as criteria for content analysis of Computer Mediated Communication (CMC). Participation rate relates directly to the raw number and timing of messages. Our definition of *interaction* is however, not the same as that of Henri (1992). Her model consist of three steps, (i) "communication of information" (ii) a first response to this information and (iii) a second answer related to the first. Interaction in our study is much more complex, in as much as we have eight (including the tutor) participants, and we have weighted our coding decisions for delivering artefacts, offering alternative solutions to group problems, and, active engagement within the online environment highly in comparison to the rest of our scoring categories. We are therefore more comfortable in defining *interaction* in line with Hara, Bonk & Angeli (2000), where we have identified messages by one of three categories: explicit, implicit and independent. During the discussion between the two raters of the transcript, the raters will work on the basis of these three category, we however, have not adopted the method used for the same by Hara, Bonk & Angeli (2000).

5 TRANSCRIPT RELIABILITY CLEANING PERCENTAGE (TRCP)

In this section, we describe what we mean by Transcript Reliability Cleaning Percentage (TRCP) using message unit. Table 2 above has only been coded using the criteria set out in Table 1 by the author. It is not possible to fully rely on coded transcripts as presented in Table 2 as it can be seen as being very subjective, and that the author, as the owner of the "problem situation" could have misinterpreted or over rated some of the messages for his own reasons in order to end up with a particular outcome. TRCP will involve an independent rater, who will also have access to the details in Table 1, as well as the same transcript used to generate Table 2. It is the job of the independent rater to seek clarification from the original rater with respect to the rationale behind the categories of message ratings, and to fully understand the original rater's intentions before generating his/her own set of ratings. It is not the duty of the independent rater to convince the original rater to change his/her mind about the coding decisions. Once the independent rater is satisfied that he/she understands the intentions behind each coding decision, the transcript should then be rated by the independent rater on their own, generating their equivalent of Table 2 above. The percentage agreement Holsti (1969) between the two coders should be calculated. If the initial percentage agreement is greater than or equal 70%, the transcript is deemed to be "clean", in this case the initial TRCP is the same as the final TRCP, otherwise, a final TRCP should be calculated before the transcript is considered to be "clean" and adequate given the subjectivity of such scoring criteria. The kappa value (Capozzoli, McSweeney, & Sinha, 1999) should be calculated from the clean transcript with a final TRCP. The kappa value for Table 3 is 0.62.

5.1 Coded Transcript with Transcript Reliability Cleaning Percentage (TRCP)

This section shows detailed analysis of the 141 postings as coded by the author. It is worth mentioning at this point that the person who was responsible for formulating Table 1 or the instigator of any online content analysis should be seen as the "expert" and their judgement ought to be deemed sufficiently adequate to interpret the final data resulting from such investigation and should be able to assist coders of the transcripts in the case where percentage agreement needs to be discussed before finalising the interpretation of results. In the case of coding protocols that include several categories, it is possible that coders may not agree on interpretations, this is why some researchers (Potter & Levine-Donnerstein, 1999) argued that although kappa is a powerful measure, it is overly conservative. Table 2 below shows the coded transcripts of the posted messages by the 7 students and their tutor. Student 1 posted 20 messages, Student 2 posted 38 messages, Student 3 posted 15 messages, Student 4 posted 22 messages, Student 5 posted 14 messages, Student 6 posted 8 messages, Student 7 posted 11messages and the tutor posted 13 messages. In order to calculate the initial Transcript Reliability Cleaning Percentage TRCP, it is recommended that a second rater is involved (see below for explanation). The initial TRCP is the percentage agreement of the two raters before discussion.

Table 2: Coded Online Message Transcripts with initial TRCP of 13%

Student 1	Student 2	Student 3	Student 4	Student 5	Student 6	Student 7	Lecturer
0,8,8,	0,8,5,	0,5,6,	0,5,8,	3,6,6,	8,6,6,	0,8,6,	7,5,4,
8,8,5	7,3,5,	5,1,7,	8,1,6,	6,4,8,	6,2,8,	6,6,6,	5,1,2,
8,5,6,	6,5,2	5,7,6,	5,5,6,	6,6,6,	5,5,	6,6,6,	8,5,2,
6,6,6,	5,5,8,	1,5,6,	6,6,6,	6,2,5,		6,6	1,8,2,
5,5,6,	5,5,6,	6,4,5,	6,6,1,	6,6			8
6,4,6,	5,8,5,		2,2,6,				
5,5	8,5,5,		4,8,5,				
	5,4,6,		2				
	6,2,8,						
	5,8,8,						
	0,6,8,						
	5,5,6						
	5,5						
Total =20	Total =38	Total = 15	Total = 22	Total = 14	Total = 8	Total = 11	Total = 13
Rating = 6	Rating = 5	Rating = 5	Rating = 5	Rating = 5	Rating = 6	Rating = 6	Rating = 4
HLE	MLE	MLE	MLE	MLE	HLE	HLE	MLE

The final rating for each student and their tutor has been rounded to the nearest whole number. This coded online message transcript is not yet "clean". This is because its Transcript Reliability Cleaning Percentage (TRCP) is currently 13% and it is not the subject of a discussion between the first and second raters yet. Table 3 shows the agreed final transcript by the two raters after discussion, with a final TRCP of 72%. Table 4 below is the participants' final rating by level of engagement in online discourse, based on the case study for this paper.

Table 3: Coded Online Message Transcripts with final TRCP of 72%

Student 1	Student 2	Student 3	Student 4	Student 5	Student 6	Student 7	Lecturer
0,8,5,	0,8,5,	0,0,6,	0,6,8,	0,6,6,	8,6,6,	0,8,6,	4,4,4,
5,8,5,	7,0,4,	5,1,6,	6,6,6,	6,4,8,	6,6,6,	6,6,6,	2,2,5,
8,5,6,	6,6,2,	6,6,8,	5,6,6,	6,6,6,	5,6	6,6,6,	5,5,5,
6,6,6,	5,5,8,	6,5,6,	6,6,6,	6,1,5,		6,2	0,5,4,
5,5,6,	5,5,6,	6,6,2	6,6,1,	6,6			5
6,4,6,	5,8,6,		6,2,6,				
5,5	5,2,6,		6,6,5,				
	5,4,6,		5				
	0,6,6,						
	5,5,8,						
	0,6,8,						
	5,4,6,						
	5,6						
Total =20	Total =38	Total = 15	Total = 22	Total = 14	Total = 8	Total = 11	Total = 13
Rating = 6	Rating = 5	Rating = 5	Rating = 5	Rating = 5	Rating = 6	Rating = 5	Rating = 4
HLE	MLE	MLE	MLE	MLE	HLE	MLE	MLE

Table 4: Category of Final Participants' Rating and Variables Investigated

Variables Investigated	Unit of Analysis (Message)	Final Rating Category
None	No engagement with the group	Low Level Engagement -LLE
Participation, Interaction	Agreeing with others without reasons	Low Level Engagement -LLE
Participation, Interaction	Agreeing with others with reasons	Low Level Engagement -LLE
Participation, Interaction	Referring the group to relevant websites	Medium Level Engagement-MLE
Participation, Interaction	Resolving conflicts within the group	Medium Level Engagement-MLE
Participation, Interaction	Taking a lead role in discussion	Medium Level Engagement-MLE
Participation, Interaction	Offering to deliver artefact(s)	High Level Engagement-HLE
Participation, Interaction	Offering alternative solutions to group problems	High Level Engagement-HLE
Participation, Interaction	Active engagement with the group	High Level Engagement-HLE

Expressed in mathematical notations, we obtain the following (which is basically saying the same thing as Table 4 above):

Low Level Engagement (Participation, Interaction) => $0 \leq \text{LLE} < 3$

Medium Level Engagement (Participation, Interaction) => $3 \leq \text{MLE} < 6$

High Level Engagement (Participation, Interaction) => $6 \leq \text{HLE} \leq 8$

6 INTERPRETATION OF INDIVIDUAL PARTICIPANT LEVEL OF ENGAGEMENT

In this section, we will interpret the result of individual participants with respect to the coded rating of messages as discussed and agreed by the two coders, resulting in the Table 3 with final TRCP of 72% with a kappa value (Capozzoli, McSweeney, & Sinha, 1999) of 0.62. Table 5 below will contribute to the interpretation in terms of the weighting attached to each coding decision by the two raters.

In Table 1, the coding decision (category) ratings are in the order of priority and hierarchy with respect to the two variables (participation and interaction) under investigation during the online negotiation of software requirements by participants. Furthermore, the higher weightings attached to categories 6, 7 and 8 have high degree of group coursework deliverables attached. This is because the overall group coursework has to be analysed, designed and implemented within a semester. In terms of cognition, this is what Henri (1992) declares to be "in-depth level processing" also adopted from Entwistle and Waterson (1998). The lower weightings attached to categories 0, 1 and 2 indicate the fact that participants may be making judgements without justification, stating that one shares ideas or opinions already communicated, repeating what has been said or just asking irrelevant questions. This category in terms of cognition as referenced previously is termed as "surface level processing". The middle weightings attached to categories 3, 4 and 5 reflect the fact that there is enough evidence to support that participants are actively involved in negotiation online with respect to the group coursework, and in some cases taking a lead role, or resolving potential group conflicts. In terms of cognition, we are suggesting that this category be termed "acceptable level processing".

Table 5: The Weighting of Coded Messages by Participant using the final TRCP of 72%

Participants	Coded Results by Category of Coding									
	0	1	2	3	4	5	6	7	8	TOTAL
Student 1	1	0	0	0	1	8	7	0	3	20
Student 2	4	0	2	0	3	12	11	1	5	38
Student 3	2	1	1	0	0	2	8	0	1	15
Student 4	1	1	1	0	0	3	15	0	1	22
Student 5	1	1	0	0	1	1	9	0	1	14
Student 6	0	0	0	0	0	1	6	0	1	8
Student 7	1	0	1	0	0	0	8	0	1	11
Tutor	1	0	2	0	4	6	0	0	0	13
TOTAL	11	3	7	0	9	33	64	1	13	141

6.1 Interpreting the Final Rating for Student 1

Student 1 made a total of 20 postings (14%) of the overall postings over the 95 days. Student 1 is a female. The Student 1 row in Table 5 shows the type of messages posted by this participant. Majority of this student's online contribution to the group (75%) relates directly with taking a lead role in discussion offering to deliver and delivering artefacts for the coursework. The student took actively engaged with the rest of the group online 15% of the time. It is evident from this student's statistical profile that she was more interested in getting the work done. This student's final overall rating as discussed and agreed by the two independent raters is High Level Engagement (see Table 3 & 4 respectively).

6.2 Interpreting the Final Rating for Student 2

Student 2 made a total of 38 postings (27%) of the overall postings over the 95 days. Student 1 is a female. She took active role in the online negotiation of software requirements. The Student 2 row in Table 5 shows the type of messages posted by this participant. Majority of this student's online contribution to the group (61%) relates directly with taking a lead role in discussion and offering to deliver and delivering artefacts for the coursework. The student actively engaged with the rest of the group online 13% of the time. This student appeared to be the most diligent of all the students in this case study, however, her statistical profile shows that her online contribution to the group embraced most of the categories. It is interesting to note that, although, she appeared to be the most active of all the students in this case study, her overall rating by the two independent raters is Medium Level Engagement (see Tables 3 & 4 respectively). This is because she sent a number of messages (10%) that did not engage the rest of the students in the group.

6.3 Interpreting the Final Rating for Student 3

Student 3 made a total of 15 postings (11%) of the overall postings over the 95 days. Student 3 is a male. The Student 3 row in Table 5 shows the type of messages posted by this participant. Majority of this student's online contribution to the group (67%) relates directly with taking a lead role in discussion and offering to deliver and delivering artefacts for the coursework. The student actively engaged with the rest of the group online 7% of the time. 13% of the messages sent by this student did not engage the rest of the group in any way, furthermore, another 13% of messages posted by this student only agreed with others with or without reasons. This student's overall rating by the two independent raters is Medium Level Engagement (see Tables 3 & 4 respectively).

6.4 Interpreting the Final Rating for Student 4

Student 4 made a total of 22 postings (16%) of the overall postings over the 95 days. Student 4 is a male. The Student 4 row in Table 5 shows the type of messages posted by this participant. Majority of this student's online contribution to the group (82%) relates directly with taking a lead role in discussion and offering to deliver and delivering artefacts for the coursework. The student actively engaged with the rest of the group online 5% of the time. 5% of the messages sent by this student did not engage the rest of the group in any way, furthermore, another 9% of messages posted by this student only agreed with others with or without reasons. This student's overall rating by the two independent raters is Medium Level Engagement (see Tables 3 & 4 respectively).

6.5 Interpreting the Final Rating for Student 5

Student 5 made a total of 14 postings (10%) of the overall postings over the 95 days. Student 5 is a male. The Student 5 row in Table 5 shows the type of messages posted by this participant. Majority of this student's online contribution to the group (71%) relates directly with taking a lead role in discussion and offering to deliver and delivering artefacts for the coursework. The student actively engaged with the rest of the group online 7% of the time. 7% of the messages sent by this student did not engage the rest of the group in any way, furthermore, another 7% of messages posted by this student only agreed with others without reasons. This student's overall rating by the two independent raters is Medium Level Engagement (see Tables 3 & 4 respectively).

6.6 Interpreting the Final Rating for Student 6

Student 6 made a total of 8 postings (6%) of the overall postings over the 95 days. Student 6 is a male. Majority of this student's online contribution to the group (88%) relates directly with taking a lead role in discussion and offering to deliver and delivering artefacts for the coursework. The student actively engaged with the rest of the group online 12% of the time. This student's overall rating by the two independent raters is High Level Engagement (see Tables 3 & 4 respectively).

6.7 Interpreting the Final Rating for Student 7

Student 7 made a total of 11 postings (8%) of the overall postings over the 95 days. Student is a female. The Student 7 row in Table 5 shows the type of messages posted by this participant. Majority of this student's online contribution to the group (73%) relates directly with taking a lead role in discussion and offering to deliver and delivering artefacts for the coursework. The student actively engaged with the rest of the group online 9% of the time. 9% of the messages sent by this student did not engage the rest of the group in any way, furthermore, another 9% of messages posted by this student only agreed with others without reasons. This student's overall rating by the two independent raters is Medium Level Engagement (see Tables 3 & 4 respectively).

6.8 Interpreting the Final Rating for the Tutor

The tutor is expected to play a role as a facilitator of information, as well as helping the group resolve conflict situations throughout the semester. The profile of the tutor from Table 5 suggests that he took lead role in the discussion (46%), he also spent some time resolving conflicts within the group (31% of his postings), 15% of his postings agreed with others with reasons, whilst 8% of his postings are general information and did not engage the students. The tutor's overall rating by the two independent raters is Medium Level Engagement (see Tables 3 & 4 respectively).

7 CODING THE TRANSCRIPT BY THE TWO RATERS

Quantitative content analysis of computer transcripts is time consuming. It took the author a total of 6 hours 30 minutes to compile the 141 messages into a word document as a transcript ready for content analysis. It took another 90 minutes for the author to independently rate the 141 messages as depicted in Table 2. In order to confirm the initial TRCP, a second rater, another tutor was asked to contribute to the research by becoming the second rater of the transcript. The second rater was given Table 1, and the rationale behind each coding decisions was elaborated upon at the time. It took the second rater 55minutes for the initial parse through the transcript. It took another 90 minutes to rate the transcript jointly by both raters after discussion in order to formulatethe final TRCP as shown in Table 3. In total, it took 10hours 25minutes to finalise the coded transcripts by the two raters. This is why few researchers using quantitative content analysis of computer conference transcripts, have published few results derived from a second content analysis.

8 CONCLUSIONS / FUTURE WORK

Our investigation has revealed that it is possible to measure the quality of online discourse using a variety of different research methods including content analysis technique. A number of eminent authors (Henri, 1992; Hara, Bonk & Angeli, 2000; Rourke, Anderson, Garrison & Archer, 2000; Cohen, 1960; Capozzoli, McSweeney & Sinha, 1999; Holsti, 1969; Hillman, 1999; Howell-Richarson & Mellar, 1969; McDonald, 1998; Kanuka & Anderson 1998; Newman, Webb & Cochrane, 1995; Weiss & Morrison 1998; Zhu, 1996, 1998) have suggested different instruments for measuring the quality of online discourse, including variables such as, participation, interaction, social, cognitive and metacognitive, levels of argumetation, critical thinking, understanding/correcting misunderstandings, focus groups, complexity of response etc. In order to measure these variables, they offer some unit of analysis, such as, message, thematic, sentence, proposition, paragraph and illocutionary act.

With respect to an alternative, or better reliability measure, we argue that our suggested Transcript Reliability Cleaning Percentage (TRCP), as documented in this paper, will better enhance interrater reliability (percentage agreement between coders) of the rating of online transcripts. We believe that it is not possible under certain circumstances to obtain 100% agreement between coders after discussion. However, we note that this was achieved by, Hara, Bonk & Angeli (2000). By adopting our proposed TRCP in the calculation of interrater reliability, we are suggesting that it is a form of second marking computer conference transcripts by an independent marker (not necessarily an expert in online assessment) with detailed marking scheme, in the form of agreed detailed coding decisions by either an expert in online assessments, the instigator of the transcript, or the author of the original transcript. This we believe adds to the body of knowledge in content analysis of computer conference transcripts.

This study has clearly demonstrated that it is time consuming to generate values for interater reliability, and calculating a kappa value when using content analysis of online discourse. Furthermore, it is difficult sometimes to achieve acceptable levels of interater reliability, this is the reason why Ravenscroft and Pilkington (2000), Barros and Verdejo (2000) and Duffy,Dueber, and Hawley (1998) have developed semi-structured computer conferencing systems, in which participants choose the type of contribution that they are making from a limited set of alternatives, as documented in Rourke, Anderson, Garrison and Archer (2000).

Existing cooperative systems for learning such as WebCT and BlackBoard have many statistical features/information, however, they lack the ability to measure participants' online discourse with respect to interaction, participation and cognition in a useful or meaningful way, except for the actual number of postings being counted by such systems as a representative measure of the same. These results are potentially very useful in the design of cooperate systems for learning, project management systems for online negotiations, and for software engineering students at higher education institutions working collaboratively to negotiate requirements of software systems.

Our future work will focus on a semi-structured computer conferencing system using the SQUAD approach to online messages. SQUAD is the type of messages posted online. This approach will be able to scaffold online discourse, making it a lot easier to code the messages by only one coder. Indeed, in the case of software engineering students, it is possible to incorporate the collation of the online messages within the marking scheme. This will also afford students the opportunity to appreciate their online contributions at the end of their course.

ACKNOWLEDGEMENTS

I would like to take this opportunity to acknowledge and thank Dr Romas Mikusauskas for agreeing to contribute to this paper by acting as the second rater of the transcript. Many thanks also to my students whose online discourse is the basis of this paper.

REFERENCES

Ahern, T., Peck, K. & Laycock, M. (1992). The effects of teacher discourse in computer-mediated discussion, *Journal of Educational Computing Research,* 8(3) 291-309.

Barros, B. & Verdejo, F. (2000). Analyzing student interaction process in order to improve collaboration: The DEGREE approach. *International Journal of Artificial Intelligence in Education,* 11, to appear.

Berelson, B. (1952). Content analysis in communication research. Illinois: Free Press. Boehm, B W (1988). A Spiral Model of Software Development and Enhancement, *Computer* May 1988, pp 61-72.

Boehm, B. W (1988) A Spiral Model of Software Development and Enhancement. *Computer,* May 1988, pp.61-72.

Capozzoli, M., McSweeney, L. & Sinha, D. (1999). Beyond kappa: A review of interratter agreement measures. *The Canadian Journal of Statistics,* 27(1), 3-23.

Cohen. J. (1960). A coefficient of agreement for nominal scales. *Education and Psychological Measurement,*20, 37-46.

Duffy, T., Dueber, B., & Hawley, C. (1998). Critical thinking in a distributed environment: A pedagogical base for the design of conferencing systems: In C. Bonk & K. King (Eds.). *Electronic contributors: learner-centered technologies for literacy, apprentiship, and discourse (pp. 51-78).* New Jersey: Lawrence Erlbaum Associates, Publishers.Entwistle, N. & Waterston, S. (1998). Approaches to studying and levels of processing in university students. *British Journal of Education Psychology,* 58, 258-65.

Hara, N., Bonk, C., & Angeli, C., (2000). Content analyses of on-line discussion in an applied educational Psychology course. *Instructional Science.* 28(2), 115-152.

Henri, F. (1991). Computer conferencing and content analysis. In A. Kaye (Ed.) *Collaborative learning through Computer conferencing: The Najaden papers,* (pp.117-136). London: Springer-Verlag.

Hillman, D. (1999). A new method for analysing patterns of interaction. *The American Journal of Distance Education,* 13(2), 37-47.

Holsti, O. (1969). *Content analysis for social sciences and humanities.* Don Mills: Addison-Wesley Publishing Company.

Howell-Richardson, C. & Mellar, H. (1969). A methodology for the analysis of patterns of participation within computer mediated courses. *Instructional science,* 24, 47-69.

Kanuka, H, & Anderson, T. (1998). Online social interchange, discord, and knowledge construction. *Journal of Distance Education,* 13(1) 57-75.

Marttunen, M. (1998). Learning of argumentation in face-to-face and email environments. (ERIC Document Reproduction Service, ED 422 791).

Marttunen, M. (1997). Electronic mail as a pedagogical delivery system. *Research in Higher Education,* 38(3), 345-363.

McDonald, J. (1998). Interpersonal group dynamics and development in computer conferencing: The rest of the story. In *Wisconsin Distance education Proceedings* Tri-Council Policy Statement: Ethical Conduct for Research Involving Humans. Available[online]: http://www.mrc.gc.ca/publications/publications.html

Newman, G., Webb, B. & Cochrane, C. (1995). A content analysis method to measure critical thinking in face-to-face and computer supported group learning. *Interpersonal Computing and Technology,* 3(2), 56-77

Oriogun P. K(1999). A Survey of Boehm's Work on the Spiral Model and COCOMO II - Towards Software Development Process Quality Improvement, *Software Quality Journal,* 8, 53-62 (1999), Kluwer Academic Publishers, USA.

Oriogun P. K (2002). Towards Understanding Software Requirements Capture: Experiences of Professional Students using the NIA to Support the Win-Win Spiral Model, *LTSN-ICS Electronic Journal (ITALICS),* ISSN 1473-7507, Volume 1, Issue 2, August 2002. Available[Online]: http://www.ics.ltsn.ac.uk/pub/italics/issue2/oriogun/007.html

Ravenscroft, A. & Pilkington, R. (2000). Rinvestigation by design: Developing dialogue models to support reasoning and conceptual change. *Journal of Artificial Intelligence in Education,* 11 (1) 273-298.

Potter, W. & Levine-Donnerstein, D. (1999). Rethinking validity and reliability in content analysis, *Journal of Applied Communications*

Research, 27, 258-284.

Rourke, L., Anderson, T., Garrison, R., & Archer, W. (2000). Methodological issues in the content analysis of computer conference transcripts, *International Journal of Artificial Intelligence in Education*, 11 to appear.

Weiss, R. & Morrison, G. (1998). Evaluation of a graduate seminar conducted by listserv. (ERIC Dodument Reproduction Service, ED 423 868).

Yahoogroups (2002) Available [online] http://www.yahoogroups.com

Zhu, P. (1998). Learning and mentoring: Electronic discussion in a distance learning course. In C. J. Bonk & K. S. King (Eds.), *Electronic collaborators: Learner centered technologies for literacy, apprentiship, and discourse* (pp. 233-259). Mahwah, NJ: Erlbaum.

Zhu, E. (1996). Meaning negotiation, knowledge construction, and mentoring in a distance learning course. (ERIC Document Reproduction Service, Ed 397 849).

THE AMERICAN JOURNAL OF DISTANCE EDUCATION, 17(4), 221–234

Transcript Reliability Cleaning Percentage: An Alternative Interrater Reliability Measure of Message Transcripts in Online Learning

Peter K. Oriogun and John Cook
London Metropolitan University

In this article, we extend previous work with respect to interrater reliability measure of computer-mediated conferencing and suggest coding categories relevant to problem-based learning. Calculating interrater reliability agreement by using a Transcript Reliability Cleaning Percentage (TRCP) approach is simple for academics with limited mathematical background and can provide insights into the nature of the learning process from the prospective of categorization of online discourse. TRCP enhances interrater reliability (percentage agreement between coders) of the rating of online transcripts.

Research on dialogue analysis has explored the relation between online dialogue features (e.g., roles, strategies, form, and content) and learning (Pilkington 2001). Such an analysis can provide useful insights into the nature of the learning processes from the perspective of, for example, what a speaker's intention is in a transmitted message and what the receiver perceives has been communicated by the message. However, problems can arise if one attempts to investigate specific categories or variables of the learning process—for example, participation, interaction, social elements, cognitive elements, and metacognitive elements (Henri 1992). In the case of coding protocols that include several categories, coders may not agree on interpretations. For this reason, some researchers (e.g., Potter and Le-

Requests for reprints should be sent to Peter K. Oriogun, London Metropolitan University, Department of Computing, Communications Technology and Mathematics, London North Campus, 2-16 Eden Grove, England. E-mail: p.oriogun@londonmet.ac.uk

vine-Donnerstein 1999) have argued that, although kappa (Cohen 1960) is a powerful measure, it can be overly conservative.

Consequently, in the research described in this article, two variables—participation and interaction—were investigated with the goal of generating an improved approach to interrater reliability agreement, or Transcript Reliability Cleaning Percentage (TRCP).

Commonly Measured Variables for Online Discourse

In a computer-mediated conferencing (CMC) research context, Henri's (1992) five variables of online discourse—participation; interaction; and social, cognitive, and metacognitive elements—tend to be investigated. Henri identified these five elements as key dimensions for the analysis of online discussion. She used *thematic* as a unit of analysis. McDonald (1998) used *thematic* as a unit of analysis during an investigation of six variables—the five identified by Henri, and a sixth, group development. Hara, Bonk, and Angeli (2000) used *paragraph* as a unit of analysis for the same five variables as Henri. Fahy et al. (2000) investigated interaction, participation, and critical thinking, using *sentence* as a unit of analysis. Cook (2001) used *speech act* as a unit of analysis to guide an investigation of metacognition, which was, in turn, used to assist in the development of a pedagogical agent for supporting musical composition learning when he investigated human teacher–learner interaction. Oriogun (2003) used *message* as a unit of analysis when he investigated participation and interaction.

Interrater Reliability Measure

Holsti (1969) provided the simplest and most common method of reporting interrater reliability—coefficient of reliability (*C.R.*)—as a percentage agreement statistic. The formula is

$$C.R. = 2m / n_1 - n_2$$

where: m = the number of coding decisions upon which the two coders agree
n_1 = number of coding decisions made by rater 1
n_2 = number of coding decisions made by rater 2

Cohen's kappa, on the other hand, is a statistic that assesses interjudge agreement for nominally coded data. It can be applied at both the global level (i.e., for the coding system as a whole) and the local level (i.e., for individual categories). In either case, the formula is

kappa = $(F_0 - F_C) / (N - F_C)$
where: N = the total number of judgments made by each coder (for this study 114)
F_0 = the number of judgments on which the coders agree (for this study 99)
F_C = the number of judgments for which agreement is expected by chance (for this study 15)

A number of statisticians characterize interjudge agreement as inadequate, as it does not account for chance agreement among raters (Capozzoli, McSweeney, and Sinha 1999). Therefore, with respect to Cohen's kappa, Capozzoli, McSweeney, and Sinha suggest that

> values greater than 0.75 or so may be taken to represent excellent agreement beyond chance, values below 0.40 or so may be taken to represent poor agreement beyond chance, and values between 0.40 and 0.75 may be taken to represent fair to good agreement beyond chance. (6)

In this research, we extend the work of Cohen (1960) and Capozzoli, McSweeney, and Sinha (1999) by suggesting coding categories relevant to problem-based learning (Woods 2000; Oriogun et al. 2001). We claim that calculating interrater reliability agreement by using the TRCP approach is a useful contribution because

1. It offers a definition for unclean CMC transcripts;
2. It suggests a way of cleaning CMC transcripts;
3. It is simple for academics with little mathematical background to use;
4. It can provide useful insights into the nature of the learning process from the perspective of categorization of online discourse; and
5. It is general enough to be applied in a variety of subject disciplines and to on-campus CMC and distance education.

In her work, Henri (1992) did not report interrater reliability measure. McDonald (1998) reported Cohen's kappa value. Hara, Bonk, and Angeli (2000) reported percentage agreement and coder stability. Fahy et al. (2000) reported percentage agreement. Oriogun (2003) reported TRCP and Cohen's kappa value.

Research Methods

For our study, we used grounded theory and case studies to guide our research. Strauss and Corbin (1990) suggested that grounded theory is especially useful for complex subjects or phenomena where little is yet known (as is the case in our study). This is because of the methodology's flexibility, which can cope with complex data and its continual cross-referencing, allowing for grounding of theory in the data and thus uncovering previous unknown issues. Foreman and Johnston (1999, 381–382) suggested that "case studies can be based on real events in real organizations." Case studies were "originally devised for use in medicine and law, have long been used in business and management education as a way of encouraging students to develop analytical skills as well as enhancing their practical knowledge." Consequently, the combination of grounded theory and a case study approach allows us to extend theory into an analysis of practice. Our research question for this study is as follows:

In what ways can the quality of online *participation* and *interaction* be measured?

Defining Quality in Participation and Interaction for the Study

Participation

Our definition of quality with respect to participation extends the suggestion for criteria for grading graduate-level student participation in a CMC classroom as reported in Hutton and Wiesenberg (2000). The criteria are as follows:

- Evidence of completion of readings
- Relevance: the student's comment moves the discussion forward
- Logic: the points are expressed and elaborated well
- Insight: the points reflect a creative or novel approach
- Referencing other students' notes in their own comments
- Acknowledging the work of others: agree, debate, question, synthesize, or expand
- Appropriate etiquette (no "flaming" or sexist/racist remarks)

Interaction

With respect to interaction, we define quality along the lines of Fahy (2001), where the meaning of the interaction must be something obvious and constant within the transcripts, and it reflects the interaction of the readers' knowledge and experience with the text in the message. Irrespective of what the writer intends, what the readers understand is based on the interaction between the message and the readers' experience, knowledge, and capability for understanding the topic. We have extended Fahy's definition, for the purpose of our work, by offering the following criteria for grading graduate-level student interaction in a CMC discourse:

- Low Interaction: resolving conflicts within the group
- Medium Interaction: offering alternative solutions to group problems and offering to deliver relevant artifacts for the group's common goal
- Active Interaction: delivering relevant artifacts for the group's common goal

The Study

Our case study is from a course titled Software Engineering for Computer Science that the first author teaches at the London Metropolitan University. In the 2001–02 academic year, ninety-five students completed the course. The students were split randomly into fifteen tutorial groups, and each group on average was composed of six members. Each group had a designated tutorial assistant. There were three assessments for the course: group coursework (40%), individual coursework (10%), and a final exam (50%). The group coursework also had an individual element attracting a maximum of 13% of the marks. This individual element was their online discourse.

The group we chose for this study posted 114 messages among its six students over a period of sixty-four days. The first author extracted all the messages from this group in order to investigate the quality of each student's participation and interaction using *message* (Marttunen 1997, 1998; Ahern, Peck, and Laycock 1992) as a unit of analysis, where each message is objectively identified before producing a manageable set of cases that incorporates problem-based learning (Woods 2000; Oriogun et al., 2001) activities before categorization (see Table 1).

The TRCP Approach

After carefully reading each of the 114 messages, the first author coded them (see Table 2) using the criteria set out in Table 1. Then a second (independent) rater took part. This independent rater also had access to the details in Table 1, as well as the same transcripts used to generate Table 2. For

Table 1. Coding Decisions Based on Message Ratings

Coding Decision (Category)	Rating
No engagement with the group	0
Agreeing with others without reasons	1
Agreeing with others with reasons	2
Referring the group to relevant Web sites	3
Resolving conflicts within the group	4
Taking a lead role in discussion	5
Offering to deliver artifact(s)	6
Offering alternative solutions to group problems	7
Active engagement with the group	8

Table 2. Coded Online Message Transcripts With Initial Transcript Reliability Cleaning Percentage of 39%

Student 1	Student 2	Student 3	Student 4	Student 5	Student 6
0, 5, 3,	0, 0, 8,	0,6, 7,	4, 5, 8,	3, 7, 3,	3, 2, 3,
4, 3, 5,	2, 6, 3,	5, 7, 6,	5, 2, 8,	3, 3, 8,	3, 2
3, 3, 2,	3, 3, 5,	5, 3, 3,	4, 3, 3,	7, 5, 2,	
4, 2, 5,	5, 8, 3,	2, 2, 5,	3, 2, 6,	7, 2, 2,	
6, 3, 4,	5, 3, 4,	2, 5, 3,	0, 6, 5	5, 3, 5,	
3, 3, 3	5, 2, 2,	3, 6, 2,		3, 3, 2	
	3, 5, 5,	5, 2, 6,			
	5, 2, 2,	3, 5, 4,			
	2, 3, 3,	8, 5, 6,			
	8, 2	5, 3			
Total = 18	Total = 29	Total = 29	Total = 15	Total = 18	Total = 5
Rating = 3	Rating = 4	Rating = 4	Rating = 4	Rating = 4	Rating = 3

this study, the independent rater was an administrative member of staff with no prior experience in interrating transcripts. The independent rater sought clarification from the original coder with respect to the rationale behind the categories of message ratings, and to fully understand the original coder's intention before generating her own set of ratings. It was not the duty of the independent rater to convince the original coder to change his mind about the coding decisions. Once the independent rater was satisfied that she understood the intentions behind each coding decision, she rated the transcript independently, and eventually built her own compilation of ratings before the initial TRCP was calculated (see Table 2). The percentage agreement (Holsti 1969) between the two coders was then calculated.

If the initial percentage agreement was greater than or equal to 70%, the transcript was deemed to be "clean." In this case, the initial TRCP was the same as the final TRCP. Otherwise, a final TRCP should be calculated before the transcript can be considered to be "clean" and adequate given the subjectivity of such scoring criteria. The kappa value (Cohen 1960) should be calculated from the clean transcript with a final TRCP.

Coded Transcript With TRCP

In Table 2 and Table 3, the "Total" is the number of messages posted by each student, and the "Rating" is the average rating of the messages, using the coding decision categories outlined in Table 1. The final rating of each student was rounded to the nearest whole number. A student's participation and interaction was categorized as either Low Level Engagement for a rating of 0, 1, or 2; Medium Level Engagement for a rating of 3, 4, or 5; or High Level Engagement for a rating of 6, 7, or 8.

On the basis of the categories described in Table 1, it is evident that, from the initial coding of the transcript, none of the students posted messages agreeing with others without justification. This coded online message transcript as shown in Table 2 is currently "unclean" (for the meaning and derivation of "clean" and "unclean" transcripts, see the Discussion section). Table 3 is the "clean" transcript. The TRCP of the "clean" transcript is in line with the corresponding kappa value (0.85) for the "clean" transcript.

Table 3. Coded Online Message Transcripts With Final Transcript Reliability Cleaning Percentage of 87%

Student 1	Student 2	Student 3	Student 4	Student 5	Student 6
0, 4, 3,	0, 0, 8,	0, 6, 8,	8, 8, 8,	3, 7, 3,	3, 6, 3,
8, 3, 5,	5, 7, 3,	5, 8, 6,	7, 5, 8,	5, 3, 8,	8, 2
3, 8, 2,	3, 3, 8,	7, 8, 3,	4, 3, 3,	7, 5, 2,	
7, 2, 5,	0, 7, 3,	1, 2, 7,	3, 8, 6,	7, 2, 1,	
6, 3, 4,	7, 3, 4,	1, 8, 3,	0, 7, 8	2, 3, 5,	
3, 3, 3	8, 2, 2,	3, 6, 0,		3, 3, 2	
	8, 5, 8,	8, 0, 6,			
	8, 5, 2,	3, 5, 5,			
	5, 3, 3,	8, 5, 7,			
	8, 2	5, 3			
Total =18	Total = 29	Total = 29	Total = 15	Total = 18	Total = 5
Rating = 4	Rating = 4	Rating = 5	Rating = 6	Rating = 4	Rating = 4

Students' Final Rating Values

The rating values in Table 2 and Table 3 have been categorized. The interpretation of the rater's scoring of the computer conference transcript by message unit is not the same as the rating values in Table 2 and Table 3. Each scoring coding decision has the variables *participation* and *interaction* embedded within it (see Table 4).

Examples of the Online Discourse

Table 5 shows a representative sample of the messages posted by the students during their online discourse. Five of the actual messages are shown in the Appendix.

Interpretation of Results

The graph in Figure 1 shows the final TRCP rating of the six students' online discourse in percentages. All the messages posted by Student 5 and Student 6 did engage the rest of the students. Student 3 and Student 5 were the only two who posted messages agreeing with others without reasons. All the students directed the rest of the group to relevant Web sites at some point during their online discourse. Student 1, Student 2, and Student 3 tried to resolve conflicts within the group during the semester. Only Student 6 did not take a lead role in the discussion. Student 2 and Student 5 did not offer to deliver ar-

Table 4. Category of Final Student's Rating and Variables Investigated

Variables Investigated	Unit of Analysis (Message)	Final Rating Category
None	No engagement with the group	LLE
Participation, interaction	Agreeing with others without reasons	LLE
Participation, interaction	Agreeing with others with reasons	LLE
Participation, interaction	Referring the group to relevant Web sites	MLE
Participation, interaction	Resolving conflicts within the group	MLE
Participation, interaction	Taking a lead role in discussion	MLE
Participation, interaction	Offering to deliver artifact(s)	HLE
Participation, interaction	Offering alternative solutions to group problems	HLE
Participation, interaction	Active engagement with the group	HLE

Note: LLE = Low Level Engagement; MLE = Medium Level Engagement; HLE = High Level Engagement.

Table 5. Examples of Online Discourse for the Final Transcript Reliability Cleaning Percentage (TRCP) Transcript

Message Number	Student Number	Final TRCP Rating
4	2	0
61	2	5
62	5	7
67	5	2
83	3	4

tifacts; however, they actively engaged the rest of the group. Only Student 6 failed to offer alternative solutions to group problems.

Discussion

Quantitative content analysis of computer transcripts is time consuming. The author needed three hours and forty-eight minutes to compile the 114 messages into a word processing document as a transcript ready for content analysis, and another fifty minutes to independently code the 114 messages as depicted in Table 2. Table 2 represents the "unclean" transcripts, with an initial TRCP of 39%. This is based on the percentage agreement of the two raters without discussion, regarding the transcript of the first rater (the owner of the problem situation). It took three days for the independent rater to submit a coded transcript, while fitting the task within her own work schedule. Her actual time spent was five hours and forty-five minutes.

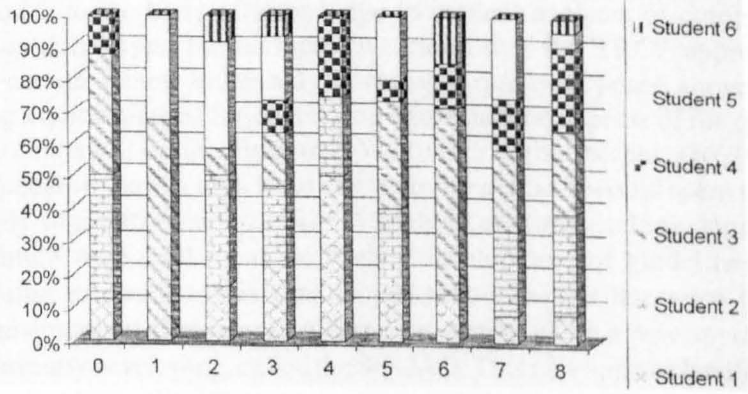


Figure 1. Graphical Representation of Category of the Final Transcript Reliability Cleaning Percentage Ratings of Students

After calculating the initial TRCP percentage for Table 2, the first author then met with the independent rater to discuss the two codings for the transcript, with the intention of further increasing the number of judgments on which they agreed. Hence, finalizing Table 3 (the "clean" transcript, with a final TRCP rating of 87%, the percentage agreement of the two raters after discussion) took forty minutes. It took eleven hours to finalize the coded transcripts by the two raters. We believe this is the reason that few researchers using quantitative content analysis of computer conference transcripts have published results derived from a second content analysis.

Conclusions

This study demonstrates that it is time consuming to generate values for interrater reliability and to calculate a kappa value when using content analysis of online discourse. Furthermore, it is sometimes difficult to achieve acceptable levels of interrater reliability. For this reason, a number of researchers have developed semistructured computer conferencing systems, in which participants choose the type of contribution that they are making from a limited set of alternatives (Rourke et al. 2001).

We argue that TRCP enhances interrater reliability (percentage agreement between coders) of the rating of online transcripts. The TRCP approach provides a form of second marking for computer conference transcripts by an independent marker (not necessarily an expert in online assessment) with a detailed marking scheme. It takes the form of an agreed detailed coding decision by either an expert in online assessments, the instigator of the transcript, or the author of the original transcript. This, we believe, adds to the body of knowledge in content analysis of computer conference transcripts. Furthermore, we claim that the TRCP approach has, to a certain extent, addressed our research question, posed above, by providing a quick method for addressing agreement on aspects of the quality of the content of online discourse. We further claim that the TRCP approach appears to have a high level of generality in that it would seem to be a relatively straightforward exercise to apply in another teaching domain.

Our future work will focus on further development of guidelines for TRCP. Future extension to this research will be to develop a theoretical basis for a semistructured computer conferencing system using a new approach we are currently developing, called the SQUAD. This new approach will be a way of scaffolding online discourse, which we believe would make the coding of online messages much easier, avoiding the interrater reliability issues when analyzing computer conference transcripts.

References

Ahern, T., K. Peck, and M. Laycock. 1992. The effects of teacher discourse in computer-mediated discussion. *Journal of Educational Computing Research* 8 (3): 291–309.

Capozzoli, M., L. McSweeney, and D. Sinha. 1999. Beyond kappa: A review of interrater agreement measures. *The Canadian Journal of Statistics* 27 (1): 3–23.

Cohen, J. 1960. A coefficient of agreement for nominal scales. *Education and Psychological Measurement* 20:37–46.

Cook, J. 2001. Bridging the gap between empirical data on open-ended tutorial interactions and computational models. *International Journal of Artificial Intelligence in Education* 12:85–99. Abstract available online at http://www.cogs.susx.ac.uk/ijaied/

Fahy, P. J. 2001. Addressing some common problems in transcript analysis. *International Review of Research in Open and Distance Learning* 1 (2). Available online at http://www.irrodl.org/content/v1.2/research.html

Fahy, P. J., G. Crawford, M. Ally, P. Cookson, V. Keller, and F. Prosser. 2000. The development and testing of a tool for analysis of computer mediated conferencing transcripts. *Alberta Journal of Education Research* 46 (1): 85–88.

Foreman, J., and T. Johnston. 1999. Key aspects of teaching and learning in business and management studies. In *A handbook for teaching and learning in higher education,* ed. H. Fry, S. Ketteridge, and S. Marshall, 372–390. London: Kogan Page.

Hara, N., C. Bonk, and C. Angeli. 2000. Content analyses of on-line discussion in an applied educational psychology course. *Instructional Science* 28 (2): 115–152.

Henri, F. 1992. Computer conferencing and content analysis. In *Collaborative learning through computer conferencing: The Najaden Papers,* ed. A. Kaye, 117–136. London: Springer-Verlag.

Holsti, O. 1969. Content analysis for social sciences and humanities. Don Mills: Addison-Wesley Publishing Company. Working Knowledge Productive Learning at Work, International Conference, The Research into Adult and Vocational Learning Group, University of Technology at Sydney, New South Wales, Australia.

Hutton, S., and F. Wiesenberg. 2000. Quality online participation: Learning in CMC classrooms. International Conference on Working Knowledge Productive Learning at Work, The Research into Adult and Vocational Learning Group, University of Technology at Sydney, New

South Wales, Australia, December. Available online at http://www.ualberta.ca/~tfenwick/ext/pubs/print/rav100.htm

Marttunen, M. 1997. Electronic mail as a pedagogical delivery system. *Research in Higher Education* 38 (3): 345–363.

———. 1998. Learning of argumentation in face-to-face and email environments. ERIC, ED 422791.

McDonald, J. 1998. Interpersonal group dynamics and development in computer conferencing: The rest of the story. In proceedings of the 14th Conference on Distance Teaching and Learning. Madison, WI: Continuing and Vocational Education, University of Wisconsin-Madison.

Oriogun, P. K. 2003. Content analysis of online inter-rater reliability using the Transcript Reliability Cleaning Percentage (TRCP): A software engineering case study. Presented at the ICEIS 2003 Conference, April 23–26, Angers, France, pp. 296–307.

Oriogun, P. K., A. W. Ferguson, M. Ludmann, N. Maisuria, and M. Swhe Yu. 2001. Using the problem-based learning grid in the management of a software engineering module: Towards an effective learning intervention for final year undergraduates. In *The power of problem-based learning*, ed. P. Little and P. Kandlbinder, 99–110. Australia: Australian Problem Based Learning Network.

Pilkington, R. 2001. Analysing educational dialogue interaction: Towards models that support learning (Introduction to the IJAIED Special Issue on Analysing Educational Dialogue). *International Journal of Artificial Intelligence in Education* 12:1–7. Available online at http://www.cogs.susx.ac.uk/ijaied/

Potter, W., and D. Levine-Donnerstein. 1999. Rethinking validity and reliability in content analysis. *Journal of Applied Communications Research* 27:258–284.

Rourke, L., T. Anderson, D. R. Garrison, and W. Archer. 2001. Methodological issues in the content analysis of computer conference transcripts. *International Journal of Artificial Intelligence in Education* 12 (1): 8–22.

Strauss, A., and J. Corbin. 1990. *Basics of qualitative research: Grounded theory procedures and techniques*. Sage: London.

Woods, D. R. 2000. Helping your students gain the most from PBL. In *Problem-based learning: Educational innovation across disciplines*, ed. T. O. Seng, P. Little, H. S. Yin, and J. Conway, 12–36. Proceedings in Conjunction with the 2nd Asia Pacific Conference on Problem-Based Learning, December 5–7, Singapore.

Appendix
Messages Sent by the Students

Message 4
From: "Student 2" <student2 email.>
Date: Wed Feb 27, 2002 1:15 pm
Subject: Testing Please Ignore

Message 61
From: "Student 2" <student 2 email>
Date: Tue Mar 26, 2002 2:40 pm
Subject: Group Meeting
Hi guys
I think we need to meet up during this holidays, well not much of a holiday for us.
Ok i was wondering if Thursday is ok with everyone, i know its very sort notice. If not next week we should diffently meet up next week. No one has any choice on that. And student 6 we want to see you there.
Meeting Time and Place: Stapleton House Café, 1.00pm – 2.00pm
Please be there guys. Please reply to this mail to confirm that you'll be there.
Regards
Student 2.

Message 62
From: "Student 5" <student 5 email>
Date: Tue Mar 26, 2002 6:36 pm
Subject: Re: Z Specification
Dear student 6,
Like the others, I am extremely impressed with the Z specification you have uploaded. It is enormous and seems very comprehensive. I do have a couple of suggestions for you to entertain, however. (1) I can't remember much Z; if you could append some explanatory text to your formal schemas, I think that would help us follow your reasoning. (2) Have you considered using Object-Z, rather than just normal Z ? I ask, because, we are using Java as our programming substrate. Java is an OO language, however, whilst Z is meant to specify imperative language programs. I fear that there may be an impedance mismatch, if you use nor-

mal Z to model our OO program. It may be, that the second point is more hypothetical than real. However, we would be naïve to rule it out without proper consideration. regards, student 5.

Message 67
From: "Student 6" <student 5 email >
Date: Tue Mar 26, 2002 7:26 pm
Subject: Re: DSDM Methodology
Dear student 2, I just finished reading your reports. I'm well impressed, well done! I really can't think of any other way in which you can improve it. ... regards, student 5.

Message 83
From: "Student 3" <student 3 email>
Date: Mon Apr 1, 2002 11:54 pm
Subject: Data Dictionary
Hi student 5, I was looking through the Data Dictionary, I see that you have amended it to clarify the potential confusion that could have arisen, which I pointed out during one of our group meetings. Thanks very much for making the amendment
Should of posted this message ages ago, just remembered tonight that I must post this message to say thanks.
Regards, student 3.

INTERACT
INTEGRATE
IMPACT

Proceedings of the 20th Annual Conference of the Australasian Society for Computers in Learning in Tertiary Education (ASCILITE)

Adelaide, Australia
7–10 December 2003

Editors
Geoffrey Crisp, Di Thiele, Ingrid Scholten, Sandra Barker, Judi Baron

Citations of works should have the following format:

Author, A. & Writer B. (2003). Paper title: What it's called. In G.Crisp, D.Thiele, I.Scholten, S.Barker and J.Baron (Eds), *Interact, Integrate, Impact: Proceedings of the 20th Annual Conference of the Australasian Society for Computers in Learning in Tertiary Education*. Adelaide, 7-10 December 2003.

ISBN CDROM 0-9751702-1-X WEB 0-9751702-2-8

ascilite
Australasian Society For Computers In Learning In Tertiary Education

Published by ASCILITE www.ascilite.org.au

USING THE ENHANCED PROBLEM BASED LEARNING GRID TO GUIDE THE DOCUMENTATION OF THE WIN-WIN SPIRAL MODEL

Peter K. Oriogun, Ajeet Khatri, Zaheeda Choudhry, Manish Borkhataria
Learning Technology Research Institute
Department of Computing, Communications Technology and Mathematics
London Metropolitan University, UNITED KINGDOM
p.oriogun@londonmet.ac.uk

Abstract

For this study, we used a subsection of the enhanced Problem-Based Learning Grid (ePBL Grid) as a framework for reflection during the adaptation of the Win-Win Spiral model for the development and documentation of software engineering projects at the London Metropolitan University. We claim that the ePBL Grid is a useful tool to guide the documentation required when adopting the Win-Win Spiral model. We also argue that students can benefit from using the ePBL Grid to aid the documentation of the Win-Win Spiral model when working in small teams online, on campus or off campus within higher education institutions.

Introduction

Educating Software Engineers is fundamentally based on problem solving through which students assimilate and apply knowledge and skills to problems of varying complexity, size and from diverse domains. The level of understanding of the underpinning theory and the acquired skills need to be ascertained through assessment. Traditional unseen examinations have long ceased to be recognised as the sole method of assessment. Problem-Based Learning (Boud & Filetti, 1996; Trop & Sage, 1998; Woods, 1999) and a range of associated instruments provide a vehicle for developing and enhancing different types of capabilities. In this paper we adopt the enhanced Problem-Based Learning Grid (Oriogun et al., 2002) and the guidelines for the Win-Win Spiral model (Boehm, 1996; MBASE Guidelines, 2003 Royce, 1995) to facilitate learning and knowledge acquisition, with specific reference to cognitive skills development within a software engineering environment. We present a case study from a postgraduate software engineering module at the London Metropolitan University in support of this study.

Background Information

The case study used for this paper is based on a postgraduate MSc Computing module titled Software Engineering, which the first author teaches at the London Metropolitan University. This is one of the four advanced core modules taught in the second semester of the course. In the 2002-03 academic year 28 students completed the coursework aspect of the course. There were 3 groups consisting of 6 students each, and 2 groups consisting of 5 students each. Each group had a designated tutorial assistant. The coursework represents 50% of the module overall, and the remaining 50% is the examination. The subject of our case study is the coursework component. The group we have chosen for this study consists of 6 mature students, 3 males and 3 females.

Theoretical Basis

The theoretical basis underpinning the Win-Win Spiral model and the enhanced Problem-Based Learning

Grid is explained in this section. The Win-Win spiral model extends the original Spiral model, Boehm (1988) by adding Theory-W activities to the front of each cycle. The theory argues that a project will only succeed if the critical stakeholders are all winners. Figure 1 below shows how the Win-Win spiral model adopts the Unified Process (Royce, 1995; Boehm, 1996; MBASE Guidelines, 2003) in the development of software artefacts.

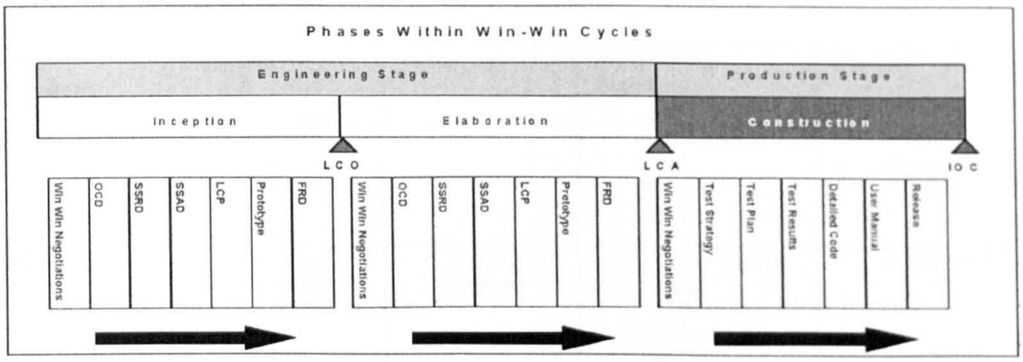


Figure 1: Anchor points within the Win-Win Cycle (Boehm, et. al. 1997)

The underpinning theoretical framework for the enhanced Problem-Based Learning Grid can be found in the model developed in Singapore as part of the national agenda for education. The "thinking programme" (Oon Seng, 2000), commonly known as the "Cognitive Modifiability Intervention (CMI)" was developed in order to enhance the ability for students to learn; to manage the learning process; for the development of students problem solving abilities and to afford students the ability to adapt to changing environment. The "Divergent-Creative Thinking Cluster" Oon Seng (2000;p.50) involved modules in developing creativity, referred to as "Problem Based Creative Learning". The model starts from "The Problem", then on to what is termed as the "Learning Adventure", followed by "Discovery Analysis and Solution Development" which leads on to "Solution, Reflection, Refinement Cycle". The "Problem Based Creative Learning" is broadly based on Problem Based Learning as recommended by (Bridges, 1992; Boud & Feletti, 1996; Trop & Sage, 1998). We have used the enhanced Problem-Based Learning Grid in this paper as a framework for reflection. Table 1 below shows the subsection of the ePBL Grid that we are reflecting upon in this paper. See Table 2 for the enhanced Problem-Based Learning Grid

Course Component	Multimedia Developer	Student
Online Resources	• *Project-manage* • *Design* • *Test prototype*	• *Interactivity*
Tutorials / Seminars / Workshops		• *Contribute* • *Ask questions* • *Engage in problem solving* • *Report progress*
Computer Mediated Communication	• *Set up discussion groups*	• *Participate* • *Engage* • *Contribute*
Teamwork		• *Research* • *Present results* • *Apply techniques* • *Implement software* • *Use tools* • *Participate*

Table 1: The ePBL Grid in the context of our Case Study

Research Methods

Our research method is through the use of a case study. Foreman and Johnston (1999, 381-382) suggest that, "case studies can be based on real events in real organizations." Our research question is as follow:

Is it possible to use the ePBL Grid as a reflective tool within the Win-Win framework?

The Enhanced Problem-Based Learning Grid (ePBL Grid)

The aim of the enhanced Problem-Based Learning Grid is to provide a structured representation of the kinds of activities undertaken by teaching and learning agents (Lecturer, Tutor, Multimedia Developer, Student - See Table 2) in order to facilitate the development of new courses that include problem-based learning as part of their pedagogical model.

Course Component	Lecturer	Tutor	Multimedia Developer	Student
Lectures	• *Plan* • *Schedule* • *Liaise with tutors* • *Deliver*			• *Attend* • *Participate*
Online Resources	• *Liaise with developer* • *Design input* • *Provide content*	• *Facilitate* • *Support*	• *Project-manage* • *Design* • *Give pedagogical advice* • *Production of assets* • *Test prototype*	• *Interactivity*
Tutorials / Seminars / Workshops	• *Plan* • *Schedule* • *Allocate*	• *Organise* • *Facilitate* • *Monitor progress* • *Liaise with lecturer*		• *Contribute* • *Ask questions* • *Engage in problem solving* • *Report progress* • *Criticize (Peer)*
Computer Mediated Communication	• *Moderate and contribute* • *Set tasks*	• *Moderate and contribute* • *Set tasks*	• *Set up discussion groups* • *Set up chat rooms* • *Set up video conference*	• *Participate* • *Engage* • *Contribute*
Research	• *Suggest* • *Monitor* • *Evaluate*	• *Suggest* • *Help* • *Focus*		• *Plan activity* • *Research* • *Investigate* • *Document*
Individual Assignment	• *Specify* • *Monitor* • *Evaluate*	• *Direct* • *Monitor*		• *Research* • *Document* • *Present* • *Implement*
Teamwork	• *Specify task* • *Allocate groups* • *Liaise with tutors* • *Evaluate results*	• *Monitor* • *Assess Progress* • *Provide Feedback*		• *Research* • *Present results* • *Apply techniques* • *Implement software* • *Use tools* • *Participate* • *Deliver presentation* • *Peer review*
Formative Feedback	• *Liaise with developer* • *Provide content* • *Provide feedback*	• *Liaise with developer* • *Provide content* • *Provide feedback*	• *Design /develop online materials* • *Produce templates*	• *Self assessment* • *Self diagnosis* • *Peer assessment*
Summative Assessment	• *Plan* • *Write* • *Deliver*	• *Support revision*	• *Design /develop online materials* • *Technical support on security issues*	• *Prepare* • *Revise* • *Attend* • *Succeed*

Table 2: The enhanced Problem Based Learning Grid - ePBL Grid (Oriogun et al., 2002)

The Study

For our study, we used the Win-Win spiral model, which follows the Unified Process (Royce, 1995, p.127). The Win-Win framework adopts the Model-Based Software Architectures (Royce, 1995), which consists of four major milestones, namely, Life Cycle Objectives (LCO), Life Cycle Architecture (LCA), Initial Operational Capability (IOC), and Product Release Milestone (PRM). These milestones map directly onto the phases within the Unified Process and are achieved at the end of each phase - see Figure 1 above. This study focuses on the subsection of the ePBL Grid that we are reflecting upon as shown in Table 1 above.

We are concentrating on the first three milestones for this study. In the first part of the study, we will look at the Life Cycle Objectives (LCO). This section will cover the setting up of discussion groups, interactivity, including students contributions, participation, and their engagement by either asking questions within the group, answering questions, offering to deliver artefacts for the groups common goal, delivering relevant artefacts for the groups common goal or generally being active members of the group through the group's win-win negotiations; this will be followed by the LCA, and, finally, the IOC. For the IOC, we will present one of our 'Administrator' Use Case diagram and one of the Class diagrams (Stevens with Pooley, 2000, p.113) for the same Use Case diagram as agreed by the group after the win conditions of each stakeholder have been met during the group's win-win negotiations (see Appendix). A snapshot of our 'Administrator' interface will be presented as implemented by the group (see Figure 3).

Life Cycle Objective for the Bulletin Board System

The LCO looks at the setting up of discussion groups by the students, the interactivity amongst the students, the online contributions made by each student, the level of engagement of each student and the overall participation of each student throughout the semester on the module. Table 3 below shows the aspect of the ePBL Grid being considered here:

Course Component	Multimedia Developer	Student
Online Resources		• *Interactivity*
Tutorials / Seminars / Workshops		• *Contribute* • *Engage in problem solving*
Computer Mediated Communication	• *Set up discussion groups*	• *Participate* • *Engage* • *Contribute*
Teamwork		• *Participate*

Table 3: Using the ePBL Grid to Reflect on the LCO

The group generated a number of documents in line with the Win-Win spiral model and Model-Based Software Architectures -see Figure 2. The group's Win-Win negotiations involved group members role-playing a number of key stakeholders including Clients, Developers, Administrators and Users of the Bulletin Board System (BBS) during the course of the semester. Each negotiation cycle involved discussion around all the sub-elements within the LCO. Conflicts were identified and if possible were resolved at this stage otherwise they were left for further negotiations in the LCA phase. The following table shows the second cycle of our negotiated win conditions with identified unresolved conflict involving all the stakeholders. Table 4 below shows how group members (stakeholders) participated, contributed, engaged in the discussion and generally interacted with the group during the development of the BBS.

Condition	Priority	Conflict		Result
		With	Details	
Internet based application	High	None	-	Agreed
User friendly system (good navigation)	High	None	-	Agreed
Full access for uploading files and messages	High	1. Operator 2. Client	1. Difficult to maintain for operator 2. Abuse, misuse and Security problems may occur.	Conflict resolved: For security reasons it cannot be allowed. Unregistered User will therefore have read-only access to messages and files.
Full access for reading/downloading files and messages	High	Client	Client suggests access only to registered users	Conflict resolved: Unregistered User will be given read-only access to all messages and files.
Search Engine	Moderate	Developers	Lack of time.	Unresolved
Help Section	Moderate	-	-	Agreed
FAQ Section	Moderate	-	-	Agreed
Suggestion Box	Moderate	-	-	Agreed

Table 4: Conflict Identification and Resolution - Win-Win Negotiations Cycle 2

Life Cycle Architecture for the Bulletin Board System

The LCA looks at the project management and the design aspects of the ePBL Grid in particular, also each student's contribution and participation individually and collaboratively online as well as off line. This is where the research element of the ePBL Grid has a role to play, students had to apply a specific technique and use appropriate case tools for the delivery of various artefacts for the coursework. Table 5 below shows the aspect of the ePBL Grid being considered here:

Course Component	Multimedia Developer	Student
Online Resources	• *Project-manage* • *Design*	• *Interactivity*
Tutorials / Seminars / Workshops		• *Contribute* • *Ask questions* • *Engage in problem solving*
Computer Mediated Communication		• *Participate* • *Engage* • *Contribute*
Teamwork		• *Research* • *Apply techniques* • *Use tools* • *Participate*

Table 5: Using the ePBL Grid to Reflect on the LCA

The Win-Win process is modelled using four main objectives (Boehm et al., 1998), Win Condition, Issue, Option and Agreement. The reconciliation phase attempts to resolve conflicts between win conditions. If a win condition is non-controversial (there is no conflict), it is covered by an agreement (Ag). Relationship between win conditions are established, leading to issues (I) being identified which raise the conflicts between win conditions and their associated risks and uncertainties. Options (Op) are considered which suggest strategies for resolving issues, which lead to agreements (Ag) that satisfy stakeholders win conditions and also define the systems objectives.

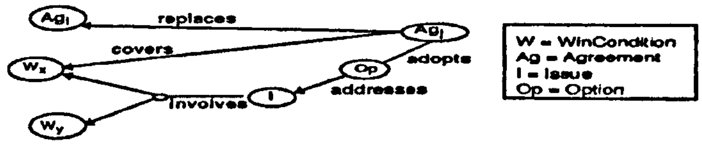


Figure 2: Win-Win decision objects and relations between them (Boehm et al., 1998)

Unresolved conflicts within the LCO were renegotiated during the LCA in order to reconcile Win-Win conditions for all the stakeholders of the BBS. The resulting agreed negotiated Win conditions are shown in Table 6 below. The difference between Table 3 and Table 4 is that it was felt by the stakeholders acting as Developers, that there was insufficient time to complete the implementation of the project, in particular the advanced features such as the search engine facility. During the cycle 3 negotiations of the LCA the group was granted an extra two weeks to complete the project. All parties were happy with implementation issues as documented in Table 6 below. Documents produced in the LCO stage were further refined in the LCA phase. The chosen architecture was further negotiated taking into consideration the analysis and design aspects of the BBS.

Condition	Priority	Conflict		Result
		With	Details	
Internet based application	High	None	-	Agreed
User friendly system (good navigation)	High	None	-	Agreed
Read-Only access to files and messages	High	None	-	Agreed
Search Engine	Moderate	Developers	Lack of time.	Conflict resolved: Two weeks extension approved.
Help Section	Moderate	-	-	Agreed
FAQ Section	Moderate	-	-	Agreed
Suggestion Box	Moderate	-	-	Agreed

Table 6: Conflict Identification and Resolution - Win-Win Negotiations Cycle 3

Initial Operational Capability for the Bulletin Board System

The IOC looks at the implementation of the software for the coursework as a prototype, the software is tested and the results are presented together with a group report documenting all the stages within the Win-Win Spiral. Table 7 below shows the aspect of the ePBL Grid being considered here:

Course Component	Multimedia Developer	Student
Online Resources	• *Test prototype*	
Tutorials / Seminars / Workshops		• *Report progress*
Teamwork		• *Implement software* • *Present results*

Table 7: Using the ePBL Grid to Reflect on the IOC

A number of documents were generated at the IOC phase in accordance with the Win-Win spiral model and Model-Based Software Architectures -see also Figure 1. The IOC is basically the implementation and the testing aspects of the Model-Based Software Architectures. For this paper, we will show a snapshot of the 'Administrator' interface and the test cases for the same interface of our BBS to be consistent with the previously illustrated Use Case and Class diagrams within the LCA. In Figure 3 below, 'Handle

Discussion' and 'Handle Files' buttons represent the 'Message Panel' and the 'Files Section' as depicted in The Appendix.

The website address for the main interface is http://simt.unl.ac.uk:9100/akk030/cctm/index.html. For the admin interface the user will require a valid username and password, which is *admin* and *cctmbbs* respectively, and can be accessed at http://simt.unl.ac.uk:9100/akk030/cctm/admin.html. During the IOC, the group conducted a total of 30 test cases for the BBS using the bottom-up software testing strategy approach (Pressman, 2000 p478) together with Black-Box testing technique (Pressman, 2000 p448). Table 8 below shows the 13 test cases performed specifically for the 'Administrator' interface.

Test Case No	Test Details	Expected Output	Test Outcome
1	Administrator login.	Welcome message - verifies login	Successful.
2	Block User.	Confirmation of user blocked.	Successful.
3	User from Test 2 logs on.	Appropriate validation message displayed	Successful.
4	Unblock User.	Message confirms user unblocked.	Successful.
5	Log on as User from result of Test 4.	Welcome message, login successful.	Successful.
6	View user details.	Table of users displayed	Successful.
7	View suggestions.	Table of suggestions displayed.	Successful.
8	Delete messages.	Message status update	Successful.
9	Archive messages.	Message status update	Successful.
10	Retrieve archived messages	Message status update from Test Result 9	Successful.
11	View messages from result of Test 10.	Message is back in normal folder.	Successful.
12	Attempt to return message from delete folder back to normal folder without clicking any checkboxes.	Validate message appears asking for a selection to be made.	Unsuccessful
13	Delete file.	Message update indicating new status of file.	Unsuccessful

Table 8: Test cases for the 'Administrator' interface

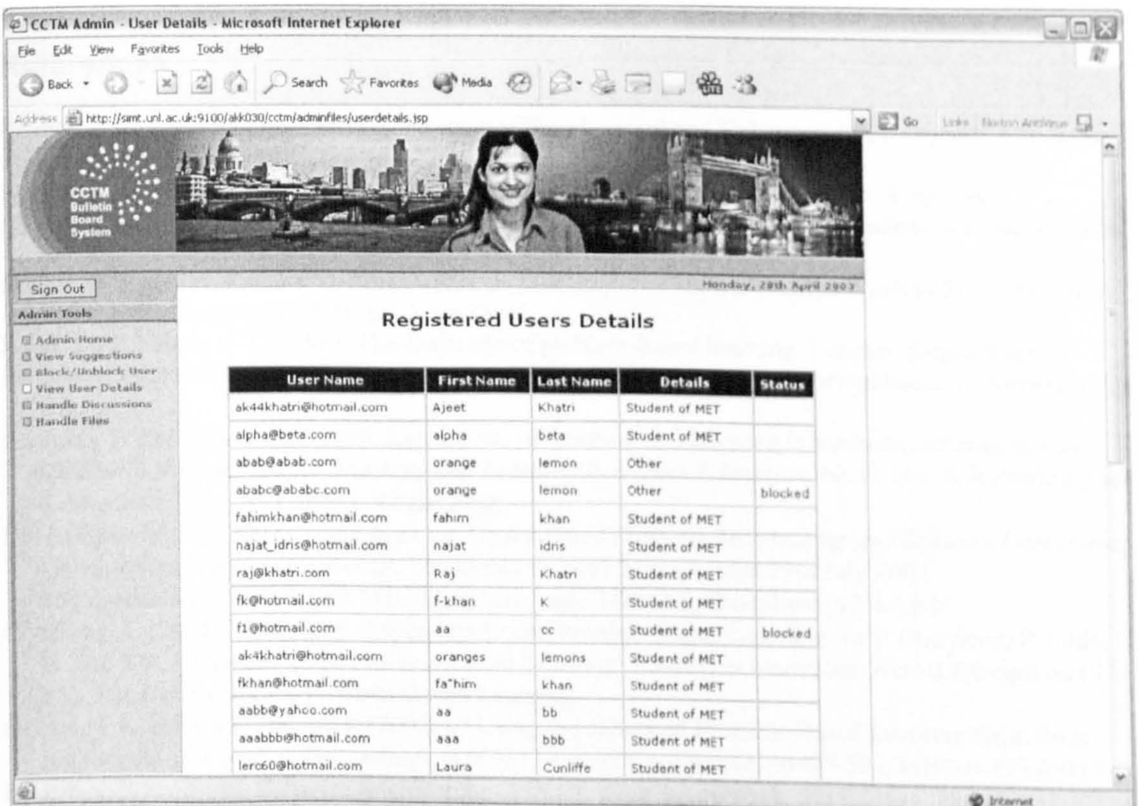


Figure 3: Snapshot of the 'View User Details'

Discussion

In a previous study (Oriogun et al., 2002) the ePBL Grid was used as a framework for reflection for three multimedia case studies. The common issues raised as a result of applying the Grid include, the use of new technology, promotion of teamwork, and working with real-life problems. It was argued in the paper that the ePBL Grid can be used to promote self-directed learning, and that it provides a structure that facilitates the logical consideration of real-life problems. It can be an invaluable aid to course design, by providing guidelines for the kind of tasks that might be suitable for learners to undertake.

From previous research (Boehm et al., 1998), it was suggested that that the results from adopting Win-Win spiral would transition well into industry, however better document guidelines are needed. Researchers at the University of Southern California are currently updating the documentation required for the Win-Win Spiral model, their version is generally known as the Model-Based (System) Architecting and Software Engineering -MBASE Guidelines (2003). In this article we argue that students can benefit from using the ePBL Grid to aid the documentation of the Win-Win Spiral model when working in small teams online, on campus or off campus.

Conclusions

We have addressed our original research question by showing that it possible to use the ePBL Grid as a reflective tool within the Win-Win framework as documented in this paper. The ePBL Grid also encourages discussion, criticism, reflection, research, peer assessment, and provides a forum that fosters interaction between students, groups of students, the use of news conferencing facilities and a forum that facilitates engagement of logical thinking to real life problems in a teaching and learning environment. We have used the ePBL Grid as a framework for reflection in documenting the Win-Win Spiral model. We claim that the ePBL Grid is a useful tool to guide the documentation required when adopting the Win-Win Spiral model.

References

Boehm, B W. (1988) A Spiral Model of Software Development and Enhancement, *Computer*, vol.21, no 5, May 1998, pp 61-72.

Boehm, B W (1996). Anchoring the Software Process, *IEEE Software*, vol. 13, no. 4, pp73-82.

Boehm B, Egyed A, Kwan J, Madachy R. (1997). Developing Multimedia Applications with the WinWin Spiral Model, Proceedings of the ESEC/FSE, 1997

Boehm, B. Egyed A, Kwan J, D. Port, A Shah & Madachy R (1998). Using the WinWin Spiral Model: A Case Study, *Computer*, vol.31, no. 7, pp 33-44.

Boud, D., & Feletti, G. I. (1996). The challenge of problem-based learning. London: Kogan Page.

Bridges, E. M. (1992). Problem-based learning for administrators. ERIC ClearingHouse. University of Oregon.

Foreman, J., and T. Johnston. 1999. Key aspects of teaching and learning in business and management studies. In *A Handbook for Teaching and Learning in Higher Education*, ed. H. Fry, S. Ketteridge, and S. Marshall, 372-390. London: Kogan Page.

MBASE Guidelines (2003). *Guidelines for Model-Based (System) Architecting and Software Engineering*, University of Southern California, [Available Online]: Downloaded 29th July 2003 http://sunset.usc.edu/research/MBASE/mbase_team/MBASE_Guidelines_v2.4.0.pdf

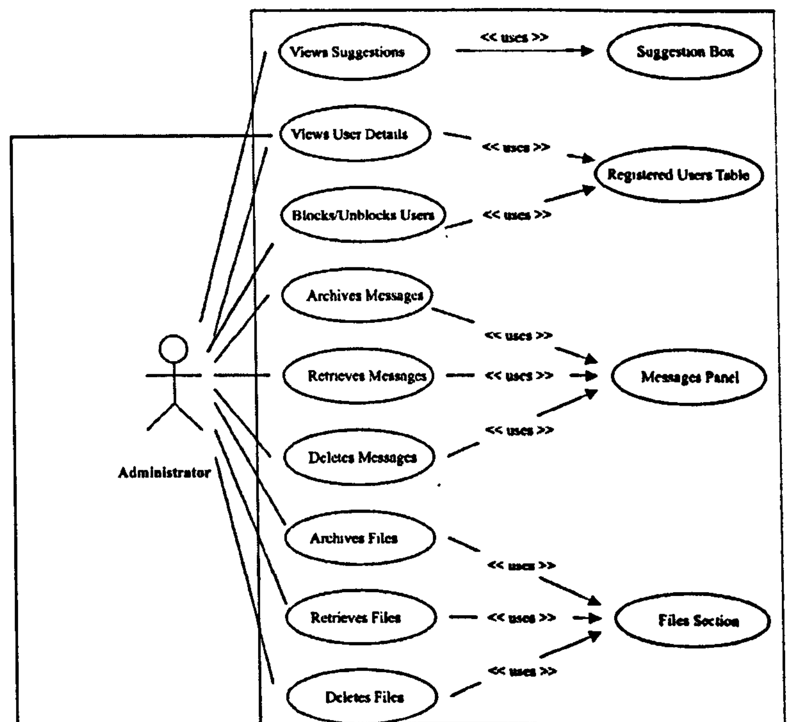
Oon Seng. T. (2000). Thinking skills, creativity and Problem-Based Learning. In T. Oon Seng, P. Little, H. Soo Yin, J. Conway (Ed), Problem-Based Learning: Education Innovation Across Disciplines (47-55): Temasek Centre for Problem-Based Learning.

Oriogun P K, French F, Haynes R (2002c) " Using the enhanced Problem-Based Learning Grid: three multimedia case studies Proceedings, ASCILITE 2002 Conference, pp495-504, ISBN 0-473-09119-4.

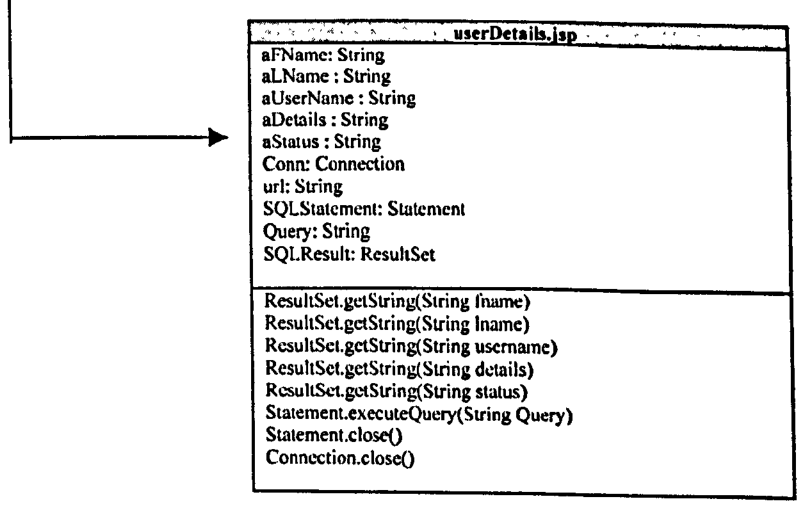
Pressman, R S. (2000). Software Engineering -A Practitioner's Approach, 5th Edition, European Adaptation, McGraw Hill.

Royce, W (1995). *Software Project Management: A Unified Framework*, Addison-Wesley Publishing Company, ISBN 0-201-30958-0, pp109-155
Stevens, P., with R. Pooley. 2000. *Using UML -Software engineering with Objects and Components*, Addison-Wesley Publishing Company, ISBN 0-201-64860-1, pp29-40.
Trop L., & Sage S. (1998). Problems as possibilities: problem-based learning for K-122 Virginia: ASDC.
Woods, D. R. (1999). Workshops. In *Problem-based learning: resources to gain the most from PBL*, (Chapters B & C). Woods, Waterdown.

Appendix: Use Case Diagram for Administrator Interface and the *Class Diagram for 'View User Details'*



Use-Case diagram for 'Administrator' Interface



Class Diagram for 'View User Details'

Acknowledgements

We would like to take this opportunity to acknowledge the efforts of Laura Cunliffe, Najat Idris, and Fahim Khan, the rest of the group members who participated in the coursework assignment.

Australian Journal of Educational Technology
2003, 19(3), 371-387.

AJET 19

Towards understanding online learning levels of engagement using the SQUAD approach to CMC discourse

Peter K. Oriogun
London Metropolitan University, UK

Recent research in content analysis has shown the difficulties of achieving acceptable levels of inter-rater reliability with CMC transcripts. This has lead to the development of semi-structured computer conferencing systems, in which participants choose the type of contribution that they are making from a limited set of alternatives. This article extends previous work with respect to semi-structured approaches to online discourse, suggesting coding categories relevant for problem based learning. The SQUAD approach to online discourse offers definitions for quality with respect to participation, interaction and cognition, when using the message as the unit of CMC transcript analysis, analysing for what the author has termed as 'online learning levels of engagement'. It is argued that the theoretical basis underpinning the SQUAD approach is beneficial for the development of teamwork and cognitive reasoning when learning in small groups, and that it is a relatively straightforward exercise to apply this approach in a different mode of study or subject area.

Introduction

In recent years we have seen the widespread adoption of computer mediated communication (CMC) in education, including extensive interest in using online communications to facilitate asynchronous dialogues, eg. online teamwork. Consequently, recent research, for example on dialogue analysis, has attempted to explore the relationship between online dialogue features (eg. roles, strategies, form and content) and learning (Pilkington, 2001). Such an analysis can provide useful insights into the nature of the learning processes from the perspective of, for example, what a speaker's intention is in a transmitted message and what the receiver perceives has been communicated by the message. However, a problem arises if we wish to investigate specific categories or variables of the learning process, eg. participation, interaction, social, cognitive and metacognitive (Henri, 1992). Specifically, if online interactions are to be transcribed and analysed using some theoretical framework, then the issue of coder interpretation at the time of coding a transcript becomes important. In the research described in this article the three variables of 'participation', 'interaction' and 'cognition' are investigated with the goal of measuring what the author has termed as 'online learning levels of engagement' using the method described as the

SQUAD approach to CMC discourse, a semi-structured way of categorising online messages. A full explanation of the meaning of this new approach and its framework is included in this article (see Table 3).

The SQUAD approach to CMC discourse adopts problem based learning (Barrows, 1996; Bridges, 1992; Oriogun et al, 2002) as an instructional method with the goal of solving real problems by:

i. Creating the atmosphere that will motivate students to learn in a group setting online;
ii. Promoting group interactions and participation over the problem to be solved by the group online;
iii. Helping learners to build up knowledge base of relevant facts about the problem to be solved online;
iv. The newly acquired knowledge is shared by the group online with the aim of solving the given problem collaboratively and collectively;
v. Delivering various artefacts leading to a solution or a number of solutions to the problem to be solved online.

Research methods

For the study described below, the author employed a combination of grounded theory and case studies to guide this research. Strauss and Corbin (1990) suggest that grounded theory is especially useful for complex subjects or phenomena where little is yet known (as is the case in our study). This is because of the methodology's flexibility, which can cope with complex data and its continual cross-referencing, which allows for grounding of theory in the data thus uncovering previous unknown issues. The SQUAD approach to CMC discourse adopts coding categories relevant to problem based learning (Barrows, 1996; Bridges, 1992; Oriogun et al, 2002) by encouraging students to develop the skills of transferring knowledge into new domains, a skill that they can carry with them throughout their professional lives. This in turn empowers them with responsibilities of managing a largely self directed learning process, as a consequence, they are better equipped and informed to accept the responsibilities of mature professional life (Brine & Shannon, 1994). Foreman and Johnston (1999, p382) suggest that, "case studies can be based on real events in real organisations" (as it is in the case study we present in support of this research). Case studies were "originally devised for use in medicine and law, have long been used in business and management education as a way of encouraging students to develop analytical skills as well as enhancing their practical knowledge" (Foreman and Johnston, 1999, p382). Consequently, the combination of grounded theory and a case study approach allows the extension of theory into an analysis of practice. The research question for this study is as follows:

> *In what ways can we measure the quality of online learning levels of engagement with respect to 'participation', 'interaction' and 'cognition'?*

Literature review

A full review of the different approaches to interaction analysis is beyond the scope of this article. Briefly, the observation and analysis of human communicative interactions are variously called content analysis, conversational analysis, discourse analysis, speech acts, dialogue analysis, and so on. Dialogue analysis is an approach that focuses on examining the patterns to be found in educational interactions. Pilkington's (1999) approach, called DISCOUNT, is based on an attempt to synthesise and extend existing approaches to discourse analysis, including transactional analysis, dialogue game theory and rhetorical structure theory. A number of researchers have developed semi-structured computer conferencing systems, in which participants choose the type of contribution from a limited set of choices. In the issue based discussion forum developed by Duffy, Dueber, and Hawley (1998), students post a message by selecting one of four labels - Hypothesis, Important Point, Evidence, or Learning Issue. Barros and Verdejo (2000) developed a system that includes automatic message analysis features.

Other approaches (eg., Cook, 2001; Baker, 1994) are based on speech act theory, (Austin, 1962; Searle, 1969). In 'classical' speech act theory, only invented, isolated utterances are considered. More recently in speech act theory, dialogue is viewed as a sequence of speech acts, uttered by each party to achieve certain goals. For example, Cook (2001) presents an approach to using empirical data on human teacher-learner interactions to guide the development of a pedagogical agent for supporting musical composition learning. In addition, dialogue analysis is also used to ascertain whether or not a user interacts within a CMC system in a productive manner, in terms of the learning task outcome. This latter approach is the focus of the work described in this article. The SQUAD approach to CMC discourse invites students to post messages based on five given categories, namely, *Suggestion, Question, Unclassified, Answer and Delivery.*

Commonly measured variables for online discourse

This section explores the literature on the variables used for content analysis of online discourse. Five variables that tend to be investigated in a CMC research context: participation, interaction, social, cognitive and metacognitive elements of online discourse. For example, Henri (1992), identified these five elements as key dimensions for the analysis of online discussion. She used *thematic* as a unit of analysis. Weiss and Morrison (1998) investigated critical thinking, understanding/correcting, misunderstanding and emotion using *thematic* and *message* as units of analysis. McDonald (1998) used *thematic* as a unit of analysis during the investigation of six variables, namely, participation, interaction, group development, social, cognitive and metacognitive elements.

By contrast, Howell-Richardson and Mellar (1996) used *illocutionary act* (from speech act theory) as a unit of analysis when they investigated participation, illocutionary properties and focus groups. Hara, Bonk and Angeli (2000) used *paragraph* as a unit of analysis for the same five variables as Henri (1992). Fahy et al. (2000) investigated interaction, participation and critical thinking, using *sentence* as a unit of analysis. Oriogun (2003) used *message* as a unit of analysis when he investigated participation and interaction.

Defining participation indicators for the 'SQUAD' approach

The three variables investigated in this article using the SQUAD approach to online messaging in this study are *participation, interaction* and *cognition.* With respect to this study, the author will first define *participation,* followed by *interaction,* and, finally, *cognition,* with a supporting theoretical basis for the three variables. The definition of quality with respect to *participation* extends the suggestion for grading graduate level student participation in CMC classroom as reported in Hutton and Wiesenberg (2000). The criteria are as follows:

- Evidence of completion of readings
- Relevance: the student's comment moves the discussion forward
- Logic: the points are expressed and elaborated well
- Insight: the point reflect a creative or novel approach
- Referencing other students' notes in their own comments
- Acknowledging the work of others: agree, debate, question, synthesise, or expand
- Appropriate etiquette (no 'flaming' or sexist/racist remarks)

Defining interaction indicators for the 'SQUAD' approach

With respect to *interaction* we define quality along the lines of Fahy (2001), where the meaning of the interaction must be something obvious and constant within the transcripts, and it reflects the interaction of the reader's knowledge and experience with the text in the message. Irrespective of what the writer intends, what the readers understand is based on the interaction between the message and the readers' experience, knowledge, and capability for understanding the topic. We have extended Fahy's definition, for the purpose of this study, by offering the following criteria for grading graduate level student *interaction* in a CMC discourse:

- Low interaction: Resolving conflicts within the group
- Medium interaction: Offering alternative solutions to group problems and offering to deliver relevant artefacts for the group's common goal
- Active interaction: Delivering relevant artefacts for the group's common goal

Defining cognitive indicators for the 'SQUAD' approach

According to (Ryder, 1994) knowledge is constructed by learners as they engage in dialogue. Furthermore, since the introduction of the "Zone of Proximity Development" (ZPD) continuum by Vygotsky (1962), it has been advocated by a number of authors that social interactions can act as scaffolding in the construction of knowledge. On the basis of this Vygotskian viewpoint, learning can be seen as a social phenomenon and experience.

A number of theories on knowledge building emphasise the socially distributed nature of cognition. Distributed cognition is therefore a process whereby individual cognition is extended to acquire something that an individual would be unable to achieve alone. Knowledge is constructed in associated networks of concepts and nodes. As learning occurs, new information is collected and coupled to existing knowledge networks. New information can then be easily retrieved to solve problems, and to apply in context. Students are expected to learn about the world based on their own research and study. Students determine their "knowns" and "unknowns". They seek knowledge to address their "unknowns". They engage in collaborative learning in their small groups to work on the problems (Wee, Kek & Sim, 2001, p159).

Bruer (1993) argues that learning is quicker when students possess self motivating skills generally referred to as metacognitive skills. Learning in PBL encourages metacognitive skills. In line with our usage of PBL in this article, we have adopted the adaptation of Henri's (1992) descriptors for "Reasoning Skills" as suggested by (Hara, Bonk & Angeli, 2000) in support of cognitive indicators for the SQUAD approach. See Table 1 for these descriptors.

Table 1: Cognitive Indicators Descriptors
(adapted from Hara, Bonk & Angeli, 2000)

Reasoning skills	Definitions
Elementary clarification	Observing or studying a problem, identifying its elements, and observing their linkages in order to come to a basic understanding.
In depth clarification	Analysing and understanding a problem to come to an understanding which sheds light on the values, beliefs, and assumptions which underlie the statement of the problem.
Inferencing	Induction and deduction, admitting or proposing an idea on the basis of its link with propositions already admitted as true.
Judgement	Making decisions, statements, appreciations, evaluations and criticisms. Sizing up.
Application of strategies	Proposing coordinated actions for the application of a solution, or following through on a choice or a decision.

Online learning matrix for the SQUAD approach

In this section the author describes his proposed Online Learning Matrix (OLM) for the SQUAD approach to CMC discourse. The OLM shown in Table 2 grouped the messages posted by the students in terms of the levels of learning engagement achieved by each participant as thus:

- Very Low
- Low
- Nominal
- High
- Very High

The grouping of messages in terms of what the author considers to be the hierarchical ordering of the type of learning associated with the postings made by participants are in line with the theoretical basis that underpins the SQUAD approach as explained earlier, see also Table 3 for the cognitive indicators.

In line with the cognitive indicators underpinning the SQUAD approach (see Tables 1 and 3 respectively) the OLM was constructed. The OLM details the levels of online levels of engagements when using this new approach to categorising CMC discourse. Figure 1 is the consolidation of Table 2 in terms of «Low«, «Nominal« and «High« levels of online engagements. Consequently, messages posted with the title Question or Q, and Unclassified or U are deemed to be generally of "Low Level of Online Engagement"; messages posted with title Suggestion or S, and Delivery or D are deemed to be generally of "High Level of Online Engagement"; finally, messages posted with the title Answer or A remains as "Nominal". Therefore, depending on the level of granularity required, Figure 1 could also have been drawn showing all the five hierarchical ordering in Table 2.

Table 2: The online learning matrix (OLM) for the SQUAD approach to CMC messaging

Message Category	Very Low	Low	Nominal	High	Very High
S - Suggestion				X	
Q -Question		X			
U -Unclassified	X				
A -Answer			X		
D -Delivery					X

The SQUAD approach to online messaging - theoretical framework

Table 3 below shows the theoretical framework for the SQUAD approach to CMC discourse within a PBL environment, together with the descriptors for 'participation', 'interaction' and 'cognition'. Sections 5-7 above elaborate further on the meaning of these descriptors with respect to the theoretical basis suggested in this article in support of this new approach to online discourse.

The study

The case study presented in this article is from a module titled Software Engineering that the author teaches at the London Metropolitan University. This module is of advanced standing, and, compulsory for the MSc in Computing offering at the University. It typically attracts between 30-80 students per semester. In the 2002-03 academic year, 38 students completed the coursework element of the module in the first semester (two semester per year). The 38 students were split randomly into 7 tutorial groups, 4 groups consisting of 6 members each and 2 groups consisting of 7 members each.

Table 3: The SQUAD approach to CMC discourse:
Descriptors for participation, interaction and cognition

Message category	Description	Example	Cognitive indicators	Participation indicators	Interaction indicators
S Suggestion	The process whereby the mere presentation of an idea to a receptive individual leads to the acceptance of the idea.	Students engage with other students within their coursework groups by offering advice, a viewpoint, or an alternative viewpoint to a current one.	-Elementary classification -In depth classification -Inferencing -Judgement -Application of strategies	Students engaging other students actively by taking a lead role in online discourse by posting meaningful and relevant messages to the group.	The message will be accessed and processed by other members of the group for the cycle of communication to complete.
Q Question	A form of word address to a person in order to elicit information or evoke a response.	Students may seek clarification from the tutor or other students in order to make appropriate decisions relating to the group coursework.	-Elementary classification -In-depth classification	The message is posed in such a way that some or all the group members will engage in the ongoing discussion.	The message will be accessed and processed by other members of the group for the cycle of communication to complete.
U Unclassified	Not in the list of categories of messages stipulated by the instigator of the task at hand.	This tends to happen at the start of the online postings. Students may be unsure of what the message is suppose to convey. In most cases, it falls within one of the four classified categories.	-Elementary classification	This type of message may or may not engage other students. In most cases, the message could be re-aligned to fall within the four classified categories by the coder of the final transcript at the end of semester.	This type of message may or may not engage other students. In most cases, the message could be re-aligned to fall within the four classified categories by the coder of the final transcript at the end of semester.

A Answer	Reply, either spoken or written, as to a question, request, letter or article.	Students are expected to respond to this type of message with a range of possible solutions / alternatives.	-Elementary classification -In depth classification -Inferencing -Judgement	Responding to a query or question will inevitably involve most, if not all the group members, especially if the response is not in line with other group members' opinions.	All group members are expected to deliver parts of the final product by working collaboratively.
D Delivery	The act of distribution of goods, mail etc.	Students are expected to produce a piece of software at the end of the semester. They all have to participate in delivering aspects of the artefacts making up the software.	-Elementary classification -In-depth classification -Inferencing -Judgement -Application of strategies	Each member of the group is expected to play active role in delivering parts of the artefact making up the final software product. This is also expressed in the marking scheme for the module.	All group members are expected to deliver parts of the final product by working together collaboratively.

Each group was assigned a designated tutorial assistant. The author was the sole lecturer on the module. The tutorial assistants assisted the students during the laboratory sessions dealing with the practical aspects of the coursework for the module. There were two assessments for the course: group coursework (50%) and a final examination (50%). Group coursework had an individual element attracting 17% of the marks (this was the CMC discourse of the module). The SQUAD approach was adopted for the CMC discourse of the module throughout the semester. The author directed his students to use a previously developed, enhanced Problem Based Learning Grid (Oriogun et al., 2002) and its predecessor (Oriogun & Georgiadou, 2000) to help them to promote learning spaces as environments that provide the learner with the responsibility for their own development and allow the learner to participate and be incumbent in a new social dynamics (Oriogun et al., 2002 p497).

The five categories for the SQUAD approach were fully explained to the students at the beginning of the semester, namely that, when a student wishes to make a suggestion to the rest of the group regarding their ongoing assignment, the posting title must be Suggestion (S); in the case asking the group a specific or general question, the posting title must be Question (Q); if a student is answering a question the posting title must be Answer (A); where a student is delivering aspects of the artefacts for the group assignment the posting title must be Delivery (D); finally, the most demanding category to rate is the unclassified category. Postings leading to or with the potential of being categorised as Unclassified (U) are as follows:

i. where a student forgets to put a title or category for their posting;
ii. where a student posts two or more kinds of categories in a single message;
iii. during the data analysis stage, when some of the messages could be re-categorised by the final rater of the message transcript - in this case the author.

The author also explained the rationale and the theoretical basis for the SQUAD at the start of the semester. Students were also asked to use a publicly available bulletin board system to facilitate their CMC discourse. The reasons for this are as follows:

i. to afford students the opportunity to familiarise themselves with various commercially available bulletin board systems before selecting one;
ii. not to depend solely on the bulletin board system supported by the university;
iii. to adopt the win-win process model using an alternative discussion forum to the win-win router developed at the University of Southern California (Boehm et al. 1995).

All of the student groups decided to use the Yahoo groups system to conduct the CMC discourse. The group chosen for this study posted 237 messages among its six members over a period of 64 days. The author also participated in the group's CMC discourse. Messages 36-40 and 66 were missing. It took a total of 3hours for the author to re-categorise some of the messages posted. It is useful to read the first 20-40 messages to check that where students have titled a posting with Unclassified (U), that it is correct, otherwise the message must be re-categorised by the final rater - in this case the author. However, as this was the first time the exercise was conducted as a case study, the author checked the content of all the messages to validate that the posting titles were in the correct category. Otherwise, the posting was re-categorised accordingly.

The whole idea of the SQUAD approach is to reduce the inter-rater reliability measure of CMC discourse. It is therefore expected that once the categorisation has been explained to students fully as documented above, the only category that may be problematic is the Unclassified or U titled postings. The author re-categorised 13% of the messages posted under the category Unclassified (see more details under the results section of this article). Six of the actual messages posted by the students who participated in the study are shown in the Appendix. These were randomly selected to represent all of the categories in the SQUAD. Figure 1 below shows the contributions of the students (numbered S1 to S6) and the author during the semester.

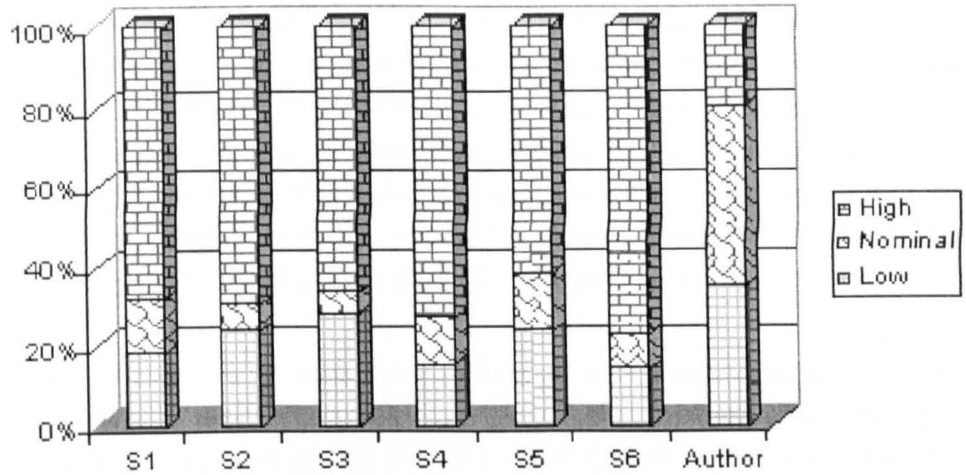


Figure 1: Participant's online learning levels of engagement (see Table 2) using the SQUAD approach to CMC discourse

Results

Student 1 sent 12.1% of all the messages posted throughout the semester. Out of the total of 28 messages sent by Student 1, 68% were of High level, 14% were of Nominal level, and 18% were of Low level. Student 2 sent 14.2% of all the messages posted throughout the semester. Out of the total of 33 messages sent by Student 2, 70% were of High level, 6% were of Nominal level, and, 24% were of Low level. Student 3 sent 15.6% of all the messages posted throughout the semester. Out of the total of 36 messages sent by Student 3, 67% were of High level, 6% were of Nominal level, and, 27% were of Low level. Student 4 sent 14.3% of all the messages posted throughout the semester. Out of the total of 33 messages sent by Student 4, 73% were of High level, 12% were of Nominal level, and 15% were of Low level. Student 5 sent 9.1% of all the messages posted throughout the semester. Out of the total of 21 messages sent by Student 5, 63% were of High level, 14% were of Nominal level, and, 23% were of Low level. Student 6 sent 15.2% of all the messages posted throughout the semester. Out of the total of 35 messages sent by Student 6, 76% were of High level, 9% were of Nominal level, and 15% were of Low level.

The author sent 19.5% of all the messages posted throughout the semester. Out of the total of 45 messages sent by the author, 20% were of High level, 44% were of Nominal level, and 36% were of Low level. The author contributed most of the category 'S', 'U', and 'A' messages. The tutor asked very few questions, and he did not post any messages under 'D' for delivery of an artefact towards the students' coursework. Figure 1 above shows a graphical representation of the students' online learning levels of engagement.

Discussion

On the whole, all the students engaged with the learning experience by contributing high levels of 'Learning Levels of Engagement' throughout the semester. All of the participants actively engaged in the learning experience as more than 60% of their CMC contributions were deemed to be of appropriate standards with respect to cognition, participation and interaction, as defined with the theoretical framework that underpins the SQUAD approach. The most interesting observation made by the author, when he adopted the SQUAD approach to online messaging, was that students were able to concentrate on the group coursework, and were able to assist one another in terms of being informative with respect to the assigned tasks. Some group members also felt that they were being watched by the author, as a consequence, they were very careful as to the tone of their messages. The group whose work is the basis of the case study in this article, pointed out in their final report that:

> Part of the project was to set up and use a yahoo groups account in order to record communication between members of the group, which was both assessed by and overseen by one of the course tutors. It was felt that the inclusion of one of the tutors created an artificial environment, partly because this meant that the developers were unable to express their feelings openly and also that some of the messages that were being posted to the site could be interpreted as seeking to impress the supervising member of staff rather than enhance the overall level of communication throughout the group.

This coursework brought the students together as they had not worked together previously within a Problem Based Learning environment at the University. The group of students further reported on their experiences as part-time students undertaking a software engineering project as quoted below:

> ...we encountered a number of problems because we are part-time evening students. One of these problems was to work on files outside of the University computing facilities. We were effectively prevented from using FTP and Telnet, which meant any testing could only be done during opening hours of the computing building at the university. These hours were very limited. Eventually we were able to overcome the problems and connect to the department of computing server, but this resulted in the loss of a week's work at least... It is a testament to the dedication of the group that we were able to meet up as frequently as was achieved, and would probably go some way to explain the dependency on using other forms of communication, including the very large number of messages sent through the yahoo groups site

The SQUAD approach is a semi-structured computer conferencing system, it is however, advisable for anyone interested in adopting the SQUAD for capturing what the author has termed as "online learning levels of engagement" to first experiment with either an inter-rater reliability measure (Cohen 1960; Henri 1992; Hara, Bonk & Angeli 2000; Fahy 2001) or the Transcript Reliability Cleaning Percentage TRCP as proposed by Oriogun (2003), in order to generate appropriate acronym from the initial categorisation of CMC messages within their subject discipline. The SQUAD approach is simple enough for academics with limited mathematical background and can provide insights into the nature of the learning process from the prospective of categorisation of online discourse.

It has been suggested by Mason & Romiszowski (1996) that

> The most glaring omission in CMC research continues to be lack of analytical techniques applied to the content of the conference transcripts. Given that the educational value of computer conferencing is much touted by enthusiasts, it is remarkable that so few evaluators are willing to tackle this research area. (p.443).

The SQUAD approach to CMC discourse is a way of addressing this suggestion by offering a theoretical framework that facilitates what the author is calling the "online learning levels of engagement".

Conclusion

The SQUAD approach adds to the significant body of literature, which argues the need for learning scaffolding when using small group learning. It is claimed that the SQUAD approach appears to have a high degree of generality in that it appears to be a relatively straightforward exercise to apply in a different mode of study or subject area. It is argued in this article that the cognitive indicators suggested for each category of message within the SQUAD approach to CMC discourse together with the participation and interaction indicators provides a quick method for addressing aspects of quality of what the author has termed as "online learning levels of engagement".

Future extension to this research will be to develop the theoretical framework to include social and metacognitive elements within a CMC message, and to develop a dedicated tool for students to use when adopting the SQUAD approach to CMC discourse.

References

Austin, J. L. (1962). *How to do things with words.* New York: Oxford University Press.

Baker, M. J. (1994). A model for negotiation in teaching-learning dialogues. *Journal of Artificial Intelligence in Education,* 5(2), 199-254.

Barrows, H. (1996). Problem-based learning in medicine and beyond: A brief overview. In L. Wilkerson and W. Gijselaers (Eds), *Bringing Problem-Based Learning to Higher Education: Theory and Practice.* New Directions for Teaching and Learning, 68, 3-11. San Francisco: Jossey-bass Publishers.

Barros, B. & Verdejo, F. (2000). Analyzing student interaction process in order to improve collaboration. The DEGREE approach. *International Journal of Artificial Intelligence in Education,* 11, 221-241. [verified 9 Nov 2003] http://computing.unn.ac.uk/staff/cgpb4/ijaied/members00/archive/vol_11/barros/full.html

Boehm, B., Bose, P., Horowitz, E. & Lee M. J. (1995). Software requirements negotiation and renegotiation aids: A theory-W based spiral approach. *IEEE Proceedings for the 17th ICSE Conference,* 1995.

Bridges, E. M. (1992). *Problem-based learning for administrators.* ERIC Clearing House, University of Oregon.

Brine, J. & Shannon, S. (1994). Consolidating professional skills and developing the confidence of graduating students. In S. E. Chen, R. M. Cowdry, A. Kingsland & M. J. Ostwald (Eds), *Reflections on Problem Based Learning.* Sydney, Australia: Wild & Wooley Pty Ltd.

Bruer, H. S. (1993). *Schools for Thought: A Science of Learning in the Classroom.* Cambridge, MA: MIT Press.

Cook, J. (2001). Bridging the gap between empirical data on open-ended tutorial interactions and computational models. *International Journal of Artificial Intelligence in Education,* 12, 85-99. [verified 27 Oct 2003] http://computing.unn.ac.uk/staff/cgpb4/ijaied/members01/archive/vol_12/cook/full.html

Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educational and Psychological Measurements,* 20, 37-46.

Duffy, T., Dueber, B. & Hawley, C. (1998). Critical thinking in a distributed environment: A pedagogical base for the design of conferencing systems. In C. Bonk & K. King (Eds). *Electronic contributors: Learner-centered technologies for literacy, apprenticeship, and discourse* (pp. 51-78). New Jersey: Erlbaum.

Fahy, P. J., Crawford, G., Ally, M., Cookson, P., Keller, V. & Prosser, F. (2000). The development and testing of a tool for analysis of computer mediated conferencing transcripts. *Alberta Journal of Education Research,* 46(1), 85-88.

Fahy, P. J. (2001). Addressing some common problems in transcript analysis, *International Review of Research in Open and Distance Learning,* 1(2). http://www.irrodl.org/content/v1.2/research.html#Fahy [viewed 24 Mar 2003, verified 18 Sep 2003]

Foreman, J. & T. Johnston. 1999. Key aspects of teaching and learning in business and management studies. In H. Fry, S. Ketteridge, and S. Marshall (Eds), *A Handbook for Teaching and Learning in Higher Education,* 372-390. London: Kogan Page.

Hara, N., Bonk, C. & Angeli, C. (2000). Content analysis of online discussion in an applied educational psychology course. *Instructional Science,* 28(2), 115-152.

Henri, F. (1992). Computer conferencing and content analysis. In A. Kaye (Ed), *Collaborative learning through computer conferencing: The Najaden papers,* pp 117-136. London: Springer-Verlag.

Howell-Richardson, C. & Mellar, H. (1996). A methodology for the analysis of patterns of participation within computer mediated courses. *Instructional Science,* 24, 47-69.

Hutton & Wiesenberg, (2000). Quality online participation: Learning in CMC classroom. *RCVET Working Knowledge Conference Papers.* Research Centre for Vocational Education and Training, University of Technology, Sydney, Australia, 10-13 Dec 2000. [viewed Mar 2003, verified 18 Sep 2003] http://www.rcvet.uts.edu.au/wkconference/working%20knowledge64.pdf

Mason, R. & Romiszowski, A. (1996). Analysing computer conferencing interactions. *Computers in Adult Education and Training,* 2(3), 161-173.

McDonald, J. (1998). Interpersonal group dynamics and development in computer conferencing: The rest of the story. In *Proceedings of 14th Annual Conference on Distance Teaching and Learning,* pp. 243-48. Madison, WI: University of Wisconsin-Madison [ERIC Document ED422864]

Oriogun, P. K. & Georgiadou, E. (2000). Towards making learning agents think of their roles in the learning process through a problem-based learning grid. *Proceedings, 2nd Asia Pacific conference on PBL,* Singapore, 4-7 Dec 2000. [verified 18 Sep 2003] http://www.tp.edu.sg/pblconference/full/OriogunGeorgiadouPaper.pdf

Oriogun, P. K., French, F. & Haynes, R. (2002). Using the enhanced Problem-Based Learning Grid: Three multimedia case studies. In A. Williamson, C. Gunn, A. Young & T. Clear (Eds), *Winds of Change in the Sea of Learning: Proceedings of the ASCILITE Conference.* Auckland, New Zealand: UNITEC Institute of Technology, 8-11 December 2002, pp495-504. http://www.ascilite.org.au/conferences/auckland02/proceedings/papers/040.pdf

Oriogun, P. K. (2003). Content analysis of online inter-rater reliability using the transcript reliability cleaning percentage: A software engineering case study. Presented at the *ICEIS 2003 Conference,* Angers, France, 23-26 April 2003, pp.296-307, ISBN 972-98816-1-8.

Pilkington, R. (2001). Analysing educational dialogue interaction: Towards models that support learning (Introduction to the IJAIED Special Issue on Analysing Educational Dialogue). *International Journal of Artificial Intelligence in Education,* 12, 1-7. [verified 9 Nov 2003] http://computing.unn.ac.uk/staff/cgpb4/ijaied/members01/archive/vol_12/pilkington/full.html

Pilkington, R. M. (1999). *Analysing Educational Discourse: The DISCOUNT Scheme* (CBLU Technical Report 99/2). Leeds: Computer Based Learning Unit, The University of Leeds, Leeds, LS2 9JT.

Ryder, M. (1994). Augmentation of the intellect: Network instruments, environments and strategies for learning. [viewed 28 Jan 2003, verified 18 Sep 2003] http://carbon.cudenver.edu/~mryder/augment.html

Searle, J. R. (1969). *Speech Acts: An essay in the philosophy of language.* Cambridge: Cambridge

University Press.

Strauss, A. and Corbin, J. (1990). *Basics of Qualitative Research: Grounded Theory Procedures and Techniques.* Sage: London.

Vygotsky, L. S. (1962). *Thoughts and Language.* Cambridge, Mass: MIT.

Wee, K. N., Kek, Y.C., & Sim H. C. (2001). Crafting effective problems for problem based learning. In P. Little & P. Kandlbinder (Eds), *The Power of Problem Based Learning,* 157-168. ISBN 0 7259 11387. Australia: Australian Problem Based Learning Network.

Weiss, R. & Morrison, G. (1998). Evaluation of a graduate seminar conducted by listserv. [ERIC Document Reproduction Service, ED 423868]

Appendix: Messages sent by students

MESSAGE 14
From: "student 2" <student2_email>
Date: Wed, 02 Oct 2002 07:26:15 +0000
Subject: S -Re: [im54p21] Re: U - Meeting

He is the customer so he would need to answer questions that you would ask any customer. It is a fairly vague specification so it is a very good idea to go back to him with a list of options.

MESSAGE 173
From: "student 1" <student1 _email>
Date: Tue Nov 19, 2002 11:15 am
Subject: Q - Formatting messages

Dear all,
This is a petty question, but...
When we paste emails into the document - should we include the original message, if a message is a reply?
I would say not, but we need to be consistent. Any thoughts?
Student 1

MESSAGE 3
From: "student 3" <student 3_email>
Date: Wed, 25 Sep 2002 22:10:11 -0000
Subject: U - [im54p21] Re: Help!

Was having similar problems earlier on, but everything seems to be OK now.
Regards
Student 3

MESSAGE 141
From: student 4 <student 4_email>
Date: Tue Nov 12, 2002 1:15 pm
Subject: A - web pages

Student 2 / student 5,
Both login and registration will have one page. The site will allow access and give message post/reply functions to both message streams (undergrad and postgrad) for any registered user.
Hope that answers it. Any chance of emailing draft html pages so I can link some code to it? Thanks
Student 4

MESSAGE 76
From: student 5 <student 5_email>
Date: Tue Oct 15, 2002 7:52 am
Subject: D - an explanation as to life cycle plans

Hi all
continuing my very busy morning, I have now put a file on with my explanation for the life cycle plan I hope that it is readable, for this time of the morning - I hope that this busy morning is taken into consideration when I am completely incoherent at this evenings meeting
can't we do genetic engineering instead ?
student 5

Author: Peter K. Oriogun
Learning Technology Research Institute
Department of Computing, Communications Technology and Mathematics
London Metropolitan University
166-220 Holloway Road, London N7 8DB
Email: p.oriogun@londonmet.ac.uk

Please cite as: Oriogun, P. K. (2003). Towards understanding online learning levels of engagement using the SQUAD approach to CMC discourse. *Australian Journal of Educational Technology*, 19(3), 371-387. http://www.ascilite.org.au/ajet/ajet19/oriogun.html

[AJET 19] [AJET home]
HTML Editor: Roger Atkinson [rjatkinson@bigpond.com]
This URL: http://www.ascilite.org.au/ajet/ajet19/oriogun.html
Created 14 Nov 2003. Last revised 14 Nov 2003.

INTRODUCING A DEDICATED PROTOTYPE APPLICATION TOOL FOR MEASURING STUDENTS' ONLINE LEARNING LEVELS OF ENGAGEMENT IN A PROBLEM-BASED LEARNING CONTEXT

Peter K. Oriogun & Eamonn Ramsay
Learning Technology Research Institute
Department of Computing, Communications Technology and Mathematics
London Metropolitan University
2-16 Eden Grove, London N7 8EA
Email: p.oriogun@londonmet.ac.uk
Tel: +44(0) 207 133 7065

ABSTRACT

Computer Mediated Communication (CMC) messaging is an invaluable tool for the promotion of learning in higher education institutions. The use of bulletin boards and email has become a standard way for lecturers to communicate with their students, and for students to communicate with one another. However one of the long-standing disadvantages regarding existing bulletin board systems is that they offer no simple way for educators to objectively measure students' levels of online learning engagement. The SQUAD approach to CMC discourse addresses the difficulty by providing a means through which statistics compiled from students' online discourse can be used to generate objective estimations of their degree of learning engagement within a problem-based learning context. In this paper the authors present SQUAD v 2.0, an enhanced implementation of the SQUAD approach, which overcomes many of the shortcomings of its predecessor. We furthermore elaborate our findings after having implemented the software within a practical learning context, and estimate its value as a means of enhancing students' online learning levels of engagement. It is our intention and hope that future work will be to develop this current version to a marketable product

KEY WORDS

Computer Mediated Communication, Problem-Based Learning, Problem-Based Learning, JavaScript, Java Servlet Pages

1. Introduction

Education increasingly requires technological tools that can support effective learning by going beyond the mere provision of knowledge and information. In addition, such tools should play a role in facilitating, guiding and managing the learning process in multiple media, information intensive environments (such as the Web). Consequently, new educational technologies should help learners to become more autonomous, to learn how to learn and reflect on their own problem-solving performance (i.e. to become life-long learners). Currently, this can be achieved through some form of communicative interaction with a human teacher (e.g. face-to-face dialogue, written feed-back, virtual interactions such as email and conferencing systems) by structuring Computer Mediated Communication (CMC) discourse between learners, and/or by providing tools that support teachers as they manage and guide learner interactions. This paper reports on the evaluation of the second in-house software prototype tool, developed within the department of Computing, Communications Technology and Mathematics, at London Metropolitan University, called SQUAD v 2.0, supporting a semi-structured approach for scaffolding student's learning online within a problem-based learning environment. The SQUAD approach [1] to CMC discourse adopts problem-based learning [2,3,4] as an instructional method with the goal of solving real problems by:

i. Creating the atmosphere that will motivate students to learn in a group setting online;

ii. Promoting group interactions and participation over the problem to be solved by the group online;

iii. Helping learners to build up knowledge base of relevant facts about the problem to be solved online;

iv. The newly acquired knowledge is shared by the group online with the aim of;

v. Solving the given problem collaboratively and collectively;

vi. Delivering various artefacts leading to a solution or a number of solutions to the problem to be solved online

Context of the Study

The evaluation presented within this paper is from a module entitled Software Engineering, which the first author teaches at London Metropolitan University. This module is of advanced standing, and is compulsory for the MSc in Computing offering at the University. It typically attracts between 30-80 students per semester (including 2 modes of study, namely full-time and part-time evening). In 2004-05, 8 students completed the coursework element of the module in the first semester (two semesters per year). All the students attended the part-time evening mode of study. The students were split into two groups, each consisting of 4 members. Most of the students have demanding full-time employment. The two groups were asked to use SQUAD v 2.0 throughout the semester to facilitate their negotiation of requirements in line with the software process model adopted for the coursework, viz. the Win-Win Spiral model [5,6,7]. The Win-Win Spiral model identifies the system and subsystem's key stakeholders; it also identifies the stakeholders' win conditions for the system or subsystem, followed by negotiating Win-Win reconciliation of the stakeholders' conditions [5,6,7].

2. The SQUAD v 2.0 Environment

The application, which supersedes [8,9] is able to calculate students 'online learning levels of engagement' [1] as well as the statistics relating to the type and number of messages posted by each student within the two groups. Only the Administrator has access to the all the SQUAD statistics, and all of the groups on the module. The Administrator interface of SQUAD v 2.0 is designed to be entirely separate from the general user section. It is implemented as a web page generated from a single Java Servlet Pages file upon the server. Client-side JavaScript is used to allow the Administrator to navigate between information pertaining to the various groups and users within the system. The Administrator can choose either to make a notice readable by a particular group or by all groups simultaneously. The main page is shown in Figure 1, and the Administrator Interface is shown in Figure 2 below. In Figure 2, we can see all the groups registered by the Administrator onto the system. A total of 10 different groups are currently on the system. The first two groups on the list are the groups whose evaluation is the basis of this paper. We will refer to them as Group 1 and Group 2 respectively.

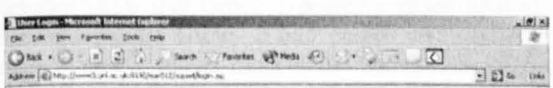


Figure 1: Main Login Page for SQUAD v 2.0

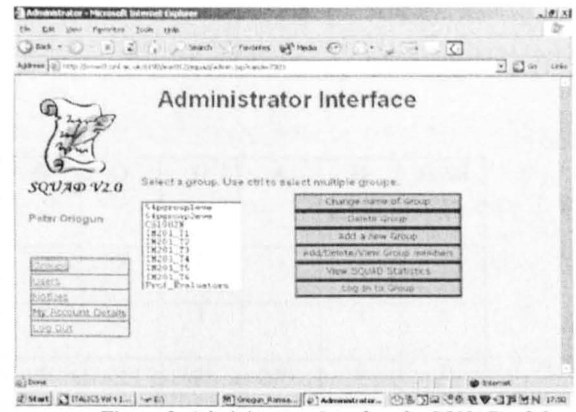


Figure 2: Administrators Interface for SQUAD v 2.0

3. Statistical Analysis of Students' CMC Posting within the SQUAD v 2.0 Environment

At the end of week 6 of semester 1 in 2004/2005 a snapshot was taken (on 15th November 2004) of the statistics generated by SQUAD v 2.0. The statistics show that Group 1 posted a total of 325 messages whilst Group 2 only posted 14 messages in total. Tables 1 shows more details about each individual student's contribution. Group 1 started with 5 members, one of the members left the module in week 5. Group 2 started with 6 members, two of the members left during week 3 and 4 of the semester. At the end of the semester Group 1 had posted 400 more messages in total than they did in week 6. Group 2 on the other hand had increased their online contributions to 143 messages.

Member Name	S	Q	U	A	D	Total	Member Name	S	Q	U	A	D	Total
Student 1	6	6	18	7	1	38	Student 5	0	0	1	1	5	7
Student 2	34	21	46	19	14	134	Student 6	0	1	1	0	3	5
Student 3	17	5	36	32	12	102	Student 7	1	0	1	0	0	2
Student 4	5	14	8	22	2	51	Student 8	0	0	0	0	0	0
Total (Group 1)	62	46	108	80	29	**325**	**Total (Group 2)**	1	1	3	1	8	**14**

Table 1: Groups & Individual Postings for Group 1 & 2 (end of week 6)

Member Name	S	Q	U	A	D	Total	Member Name	S	Q	U	A	D	Total
Student 1	18	32	55	19	13	137	Student 5	9	6	12	10	14	51
Student 2	52	36	77	31	37	233	Student 6	3	4	8	0	7	22
Student 3	44	13	89	71	32	249	Student 7	9	3	39	0	7	58
Student 4	18	24	22	36	6	106	Student 8	0	1	7	0	4	12
Total (Group 1)	132	105	243	157	88	**725**	**Total (Group 2)**	21	14	66	10	32	**143**

Table 2: Groups & Individual Postings for Group 1 & 2 (end of week 12)

At the end of week 12 of semester 1 (end of the semester) in 2004/2005 a final snapshot was taken (on 11^{th} January 2005) of the statistics generated by SQUAD v 2.0. The statistics show that Group 1 posted a grand total of 725 messages whilst Group 2 had posted 143 messages in total. Table 2 shows more detail about each individual student's contribution. The statistics shown in Table 2 depict valid data set to be used for evaluation for the whole semester.

Approximately 73% of the messages in category D – Delivery were posted by students in Group 1, compared to around 27% posted by students in Group 2 for the same message category throughout the semester. Overall, Group 1 posted just over 83% more messages than Group 2 throughout the semester. Figures 3 & 4 below represent each of the group's 'online learning levels of engagement' [1]. See Section 8 below for a detailed explanation the Online Learning Matrix, which gives the rationale for generating online learning levels of engagement. Figures 5 & 6 below shows that a particular student belongs to two different groups within the SQUAD v 2.0 environment. Figure 8 shows the interface of one of the groups to which the students belong.

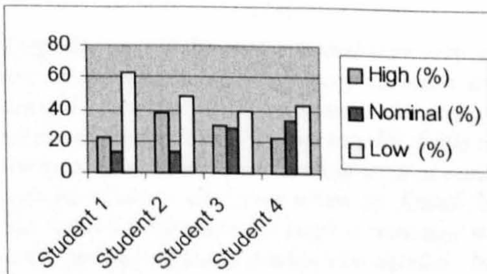


Figure 3: Graphical Representation of Table 5

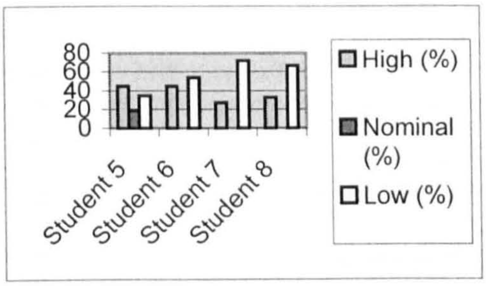


Figure 4: Graphical Representation of Table 6

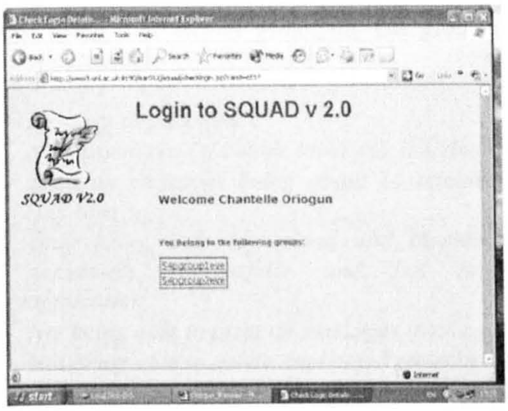


Figure 5: A Student Belonging to Two Groups

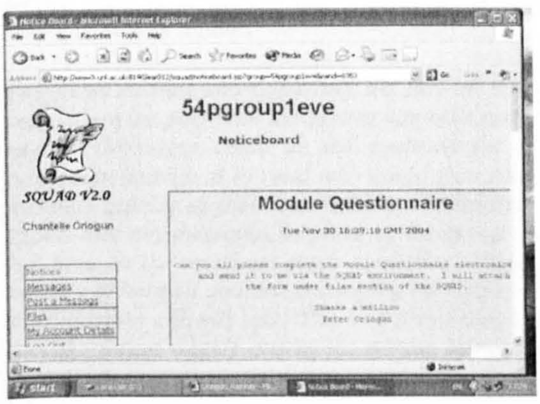


Figure 6: An Interface of one of the Student's Groups

The interface of the SQUAD v 2.0-prototype web application is split into two primary functional units. There is a general user interface, which students and ordinary users access when they log into the system (see Figure 8 below). The notice board consists of messages, which the Administrator (Lecturer, Supervisor or Super user) has left for the entire group to read. Notice board messages might include information for students about coursework deadlines, links to course related web resources and so forth. This gives SQUAD v 2.0 a distinct practical advantage over email based course communication systems, where important messages are likely to be disregarded, which can lead to organizational problems. After the student has checked the notice board he or she can navigate to the Messages section, which acts as a repository for all the messages posted by group members. The Administrator interface is slightly different from that of the ordinary user. The main reason for this is that they do not have access to SQUAD statistics and are not allowed to register or delete groups or members within groups.

4. Preliminary Evaluation of SQUAD v 2.0 by Postgraduate Software Engineering Students

As part of the deliverables for the coursework aspect of the software engineering module, the two groups of students were asked to evaluate this second prototype supporting the SQUAD approach by answering a number of questions. Only students from Group 1 evaluated this second prototype based on the questionnaire. The following excerpts are from students in Group 1. The group had not used WebCT or other notice board facility, as a group so was unable to say much about how other notice board facilities compared with this second prototype. Their evaluation of the interface design is as follows:

The design is very simple and serves its purpose in being clear to enhance navigation, but not much thought has been put into the design icon used - although the icon at least highlights one method of communication. The use of green and yellow as background colours would result in those with colour blindness finding the writing on the main page very difficult to read. *The hyperlinks in blue are easier enough to read though. forms have been well thought out and the contrast between the background and text message area will ensure that written messages can be read back before they are posted. The use of tables to display inputted text on the notice, files, account details and messages pages makes the reading of separate pages much easier to digest.*

With respect to 'ease of use', the group evaluated the prototype as follows:

Everyday use of the notice board was very simple. The use of the system was very easy to learn and navigate around. However when we came to the end of the project we found that we could not personally delete messages on the system, not even ones we had written ourselves. This was particularly annoying when we found that by using the refresh button on the page a message would repeat itself and subsequently couldn't be deleted. In addition to this nobody user or administrator was able to print all the messages on the message board in one go. Due to the volume of messages produced by the group this was a big disadvantage in terms of collation of messages for the coursework file.

On the question of what the students saw as being good about using this second prototype, the group had the following to say:

- *Able to meet online in almost real time to discuss ideas. Particularly useful for a group who could not meet face to face very often.*
- *Messages held centrally made it easier to see what people are doing.*
- *Classification methods also useful; could see when questions, answers and items were being delivered.*
- *Easy to navigate*
- *Easy to post messages, change accounts details and upload files.*

On the question of what the students saw as being bad about using this second prototype, the group had the following to say:

- *Message duplication*
- *Not easy to read if colour blind due to colour design*
- *Time on messages being about 15 minutes behind real time*
- *Only being able to upload and download word documents successfully and not any other application.*
- *Not being able to print all messages in one go.*
- *Not being able to delete duplicated messages.*

The students were asked if the SQUAD approach had helped them to participate and interact online, and their response was as follows:

Yes. Due to time shortages and the fact all students in our team were part time the system has been particularly useful. Messages could be left centrally for all or a particular individual to read who could then pick up the message quickly at their own leisure. Individuals could expect that any messages for them or the group would be left here so the messages were consistent rather than a mixture of hotmail and text messaging etc which may lead to confusions and mix ups. The board was also used as a meeting forum weekly due to the system being almost in real time.

The students were asked if they would recommend SQUAD to other students working in small teams for coursework delivery, their response were as follows:

Yes, for the advantages stated above. Particularly useful and best utilised by the group as a meeting forum. Also uploading files to a central space reduced the message delivery time for several individuals who may need the paperwork for different reasons, i.e. delivering two further sections of the coursework or using it to do some programming etc.

5. Online Learning Matrix for the SQUAD approach

The Online Learning Matrix (OLM) for the SQUAD approach to CMC discourse grouped the messages posted by the students in terms of the levels of learning engagement achieved by each participant as thus [1]:

- Very Low
- Low
- Nominal
- High
- Very High

The grouping of messages in terms of what the first author considers to be the hierarchical ordering of the type of learning associated with the postings made by participants is in line with the theoretical basis that underpins the SQUAD approach [1]. In line with the cognitive indicators underpinning the SQUAD approach the OLM was constructed. The OLM details the levels of online engagement when using this new approach to categorizing CMC discourse. Messages posted with the title Question (Q), or Unclassified (U) are deemed to be generally of "Low Level of Online Engagement"; messages posted with title Suggestion (S), or Delivery (D) are deemed to be generally of "High Level of Online Engagement"; finally, messages posted with the title Answer (A) remain as "Nominal". Therefore, depending on the level of granularity required, Figures 3 & 4 could also have been drawn showing all the five hierarchical ordering in Table 3 below.

Message Category	Very Low	Low	Nominal	High	Very High
S - Suggestion				X	
Q -Question		X			
U -Unclassified	X				
A -Answer			X		
D -Delivery					X

Table 3: The online learning matrix (OLM) for the SQUAD approach to CMC messaging (Oriogun, 2003)

6. Conclusion / Future Work

We are mindful of the fact that we only had two groups of students evaluating this second prototype. In order to validate the evaluation reported in this paper, we intend to give more software engineering students (approximately 200) the opportunity to evaluate this second version during semester 2 of 2004/2005. It is also our intention that future work will be to develop this current version to a marketable product and to secure a research funding for the project both nationally and internationally based on further evaluation of this second version.

From the statistical analysis of the two postgraduate software engineering students' online postings using the second prototype SQUAD v 2.0, it is evident that all the students engaged with the learning experience. However, it must be said that members within Group 1 were more active than those within Group 2. The sheer quantity of messages posted throughout the semester by Group 1 showed their collective enthusiasm. The group was consistent in their postings over the 12 weeks, sending approximately 55% more messages during the second half of the semester than in the first half. The group mentioned in their evaluation that initially they were driven by the marks attributed to using the message notice board; however as time progressed they actually found the prototype to be a very good method of communication. In contrast, Group 2 was motivated by the fact that marks were awarded for achieving a target level of message posting. It was to be expected that there would be many postings in the high category towards the end of the semester, as this was necessitated by the need to produce various artifacts contributing towards the final coursework.

The students felt that the design was very simple and served its purpose in being clear to enhance navigation, but they expressed unhappiness with the design of the logo. They were, however, pleased about the layout of forms and the general appearance. They reported that the system was very easy to learn and navigate, and that it was easy to post messages, upload files and meet online in almost real time. Due to the fact that all of the students were also engaged in demanding full-time employment, they found that the system was particularly useful for individual and group participation when working within a problem-based learning context, as was the case on this particular course at London Metropolitan University.

References:

[1] Oriogun P. K (2003). Towards understanding online learning levels of engagement using the SQUAD approach to CMC Discourse. Australian Journal of Educational Technology, 19(3), 371-388, ISSN 0814-673X

[2] Barrows, H. (1996). Problem-based learning in medicine and beyond: A brief overview. In L. Wilkerson and W. Gijselaers (Eds), *Bringing Problem-Based Learning to Higher Education: Theory and Practice.* New Directions for Teaching and Learning, 68, 3-11. San Francisco: Jossey-bass Publishers.

[3] Bridges, E. M. (1992). *Problem-based learning for administrators.* ERIC Clearing House, University of Oregon.

[4] Oriogun, P. K., French, F. & Haynes, R. (2002). Using the enhanced Problem-Based Learning Grid: Three multimedia case studies. *Proceedings of the ASCILITE Conference.* Auckland, New Zealand: UNITEC Institute of Technology, 8-11 December 2002, pp495-504. http://www.ascilite.org.au/conferences/auckland02/proceedings/papers/040.pdf

[5] Boehm B W, (1988). *A Spiral Model of Software Development and Enhancement*, Computer, May, 1988, pp 61-72.

[6] Boehm, B., Bose, P., Horowitz, E. & Lee M.J. (1995). Software Requirements Negotiation and Renegotiation Aids: A Theory-W Based Spiral Approach, IEEE Proceedings for the 17th ICSE Conference, 1995.

[7] Boehm B W, (2000). Spiral Development: Experience, Principles and refinements, USC Spiral Experience Workshop, 9th February 2000. http://sunset.usc.edu/MBASE

[8] Oriogun P K (2004). "SQUADgroups – A novel tool for capturing students' online learning levels of engagement with small groups", Proc ITRE 2004 , pp106-110, IEEE Publication, ISBN 0-7803-8625-6 CD-ROM 0-7803-8626-4, IEEE Catalogue Number 04EX906 CD-ROM 04EX906C.

[9] Oriogun P. K (2005). "Introducing a dedicated tool for facilitating a semi-structured approach to CMC messaging", Workshop on e-Learning Online Communities, eLOC 2005, CD-ROM, January 3-6, 2005 IEEE Catalog Number 05EX949, ISBN 0-7803-8735-X, Library of Congress Number 2004110879.

THE AMERICAN JOURNAL OF DISTANCE EDUCATION, *19*(4), 197–214

Validating an Approach to Examining Cognitive Engagement Within Online Groups

Peter K. Oriogun
Department of Computing, Communications Technology and Mathematics
London Metropolitan University

Andrew Ravenscroft and John Cook
Learning Technology Research Institute
London Metropolitan University

Tools for measuring cognitive engagement within online groups have been concerned only with measuring an individual participant's cognitive engagement, without any concern for measuring cognitive engagement within groups. There remains a serious need for a scheme that measures cognitive engagement of groups and the validation of such a scheme against existing methods. The SQUAD (coding categories that are being measured, a semistructured approach for scaffolding online groups' engagement) approach to computer-mediated communication (CMC) discourse invites students within their respective groups to post messages based on five given categories: (a) *s*uggestion, (b) *q*uestion, (c) *u*nclassified, (d) *a*nswer, and (e) *d*elivery. In this article, the authors validated the SQUAD approach at the message level with an established framework called the *practical inquiry model for assessing cognitive presence of CMC discourse.* They adopted the alignments suggested by one of the developers of the Transcript Analysis Tool at sentence level to assess students' cognitive engagement within online groups in three case studies presented in this article. The authors argue that the cognitive presence attributed to the SQUAD approach has been empirically validated with respect to cognitive engagement within groups online.

Correspondence should be sent to Peter K. Oriogun, London Metropolitan University, Department of Computing, Communications Technology and Mathematics, Learning Technology Research Institute, 2-16 Eden Grove, London N7 8EA, England. E-mail: p.oriogun@londonmet.ac.uk

The three case studies illustrate the authors' approach to negotiating and reconciling problem-solving task requirements for software engineering online. The three groups of students made effective use of all the message categories for cognitive engagement within groups online.

It has been suggested that the process of collaborative learning that occurs while learners interact to create a collective solution to a given task or problem is a form of cognitive benefit (Johnson and Johnson 1996). In such situations, learners may be encouraged to foster positive social interdependences, such as helping each other within the group to realize their potential through continuous and sustained feedback. Consequently, a collaborative, problem-based learning process can help create an atmosphere where learners are able to reflect on their own progress made within the group and in the context of a collective dedicated to completing a given task. Such a group communication medium can provide learners with the opportunity to exchange ideas related to one another and receive feedback from their peers.

One way of engaging learners in online collaborative learning is to create an environment in which knowledge emerges and is shared. The onus is therefore on the tutor/instructor to (1) create an environment in which knowledge emerges and is shared through the collaborative work within a group of students and (2) facilitate sharing of information and knowledge among members of a learning team instead of controlling the delivery and pace of course content. The SQUAD (suggestion/question/unclassified/answer/delivery) approach (Oriogun 2003b, 2005) to online discourse adopts a problem-based learning approach (Barrows 1996; Bridges 1992; Oriogun, French, and Haynes 2002) and allows groups of learners to interact for the purpose of creating a collective solution to a given task or problem and provides a way of measuring students' online learning levels of engagement (Oriogun 2003b) by

- creating the atmosphere that will motivate students to learn in a group setting online (where students are able to *trigger* a discussion within their respective groups);
- promoting group interactions and participation over the problem to be solved by the group online (where students can *explore* various possibilities within the group by actively contributing to the group);
- helping learners to build up a knowledge base of relevant facts about the problem to be solved online (where students can begin to *integrate* their ideas to influence others within their group);

- allowing the newly acquired knowledge to be shared by the group online with the aim of solving the given problem collaboratively and collectively (where students can *resolve* issues relating to the assigned work to be completed collectively); and
- delivering various artifacts leading to a solution or a number of solutions to the problem to be solved online (where students can both *integrate* and *resolve* aspects of the problem to be solved collectively).

Garrison, Anderson, and Archer's (2001) definition and use of *trigger, exploration, integration,* and *resolution* is in line with the SQUAD approach usage of these same terms. This is why we have opted to validate the SQUAD with Garrison et al.'s (2001) framework.

An examination of the existing literature to date has revealed that there are no tools for measuring the cognitive elements of groups of people working on a particular task or problem online, such as a group's coursework for a module or course. There are tools available for investigating cognitive elements of individuals working online (Fahy 2002; Garrison, Anderson, and Archer 2001; Hara, Bonk, and Angeli 2000; Henri 1992; Oriogun 2003a; Oriogun and Cook 2003). In this article, we adopt the theoretical framework of two recently developed tools, commonly used for analyzing students' cognitive elements online (Fahy 2002; Garrison, Anderson, and Archer 2000, 2001) at the individual level to validate at the group level the cognitive engagement of groups of students working within the SQUAD approach.

We adopted Fahy's (2002) suggested three different alignments of the Transcript Analysis Tool (TAT) categories with Garrison, Anderson, and Archer's (2001) model as a framework to realize the *cognitive presence* in the SQUAD approach (Oriogun 2003b, 2005). We used three case studies from three groups of master's computing students who used the SQUAD environment (software tool supporting this new approach) to negotiate and reconcile software requirements online during the two semesters of the 2003–2004 academic year at London Metropolitan University.

Each of the three case studies covered a period of twelve consecutive weeks. The first group of students posted a total of 725 messages, the second group posted 143 messages, and the third group posted 171 messages. The unit of transcript analysis for the SQUAD approach was at message level. By *message level* we mean a unit of online transcript analysis that is objectively identifiable; unlike other units of online transcript analysis, the message-level unit allows multiple coders to agree consistently on the total number of cases. It also produces a manageable set of cases. If the cognitive presence realized in this article for the SQUAD approach is accepted, using Fahy's (2002) alignments within Garrison et al.'s (2001) framework together with the case studies we present in support of our argument, we have provided a way of empirically validating Oriogun's (2003b) SQUAD approach with respect to cognitive engagement within online groups.

Cognitive Presence in Fahy et al.'s (2000) Transcripts Analysis Tool

A number of researchers have developed analytical tools for measuring online transcripts. Fahy et al. (2000) used the TAT based on Zhu's (1996) earlier work, which operates at a sentence level of analysis for the comparison of the frequencies and proportions of five categories or sentence types in a particular data set. Fahy et al.'s five coding categories are shown in Figure 1.

When Fahy (2002) examined the cognitive presence model, he realized that the categories of the TAT might be capable of being aligned with the phases in Garrison, Anderson, and Archer's (2001) model, with the resulting alignments reflecting different assumptions about the linguistic and social behavior associated with the model's phases. From three such alignments an analysis was produced, allowing a comparison of both the analytic processes involved and the resulting richness of the insights provided. In aligning the TAT with the phases of the cognitive presence model, interpretation was required. Garrison, Anderson, and Archer (2001, 14) found that elements fit multiple categories; three different alignments of the TAT categories with the model were produced, based on different assumptions about what interactive behavior is apparent in the four phases of cognition (Fahy 2002). Transcript Analysis Tool alignments with the phases of the model are shown in Table 1; also, the equivalent mapping of the SQUAD is shown in Table 3. These alignments are the basis of this article.

Cognitive Indicators in Oriogun's (2003b) Squad Approach to CMC Discourse

The SQUAD approach (Oriogun 2003b) to computer-mediated communication (CMC) discourse provides a means through which statistics compiled from students' online discourse can be used to generate objective estimations of their degree of learning engagement. The cognitive indicators of the SQUAD approach are based on Henri's (1992) cognitive indicators. The cognitive descriptors adapted from Hara, Bonk, and Angeli (2000) are shown in Table 2.

1A – vertical questions: questions which assume a 'correct' answer exists, if the right authority can be found to supply it. ['What are the categories in Bloom's taxonomy?']

1B – horizontal questions: accepts that there may not be one right answer; others are invited to help provide a plausible or alternate 'answer', or to help shed more light on the question. ['What is good teaching?']

2A – non-referential statements: contain no or very little self-revelation and usually do not invite response or dialogue; tone may be didactic; the main intent is to impart facts or information). ['Although our office has been in the business of providing program inservice and training workshops since its inception, it is new to the area of computer-mediated communications.']

2B – referential statements: postings that make direct or indirect reference to elements of preceding statements. ['I want to add to {name's} point about the importance of context in assessing technologies.']

3 – reflections: thoughts, judgments, opinions or information which are personal, or usually at least somewhat guarded or private; a tone of self-disclosure is suggested in the sharing process. ['I felt, as a teacher, that I had failed the most needy students—it's the reason I left teaching after ten years and lots of private tears.']

4 – scaffolding and engaging: intended to initiate, continue, encourage or acknowledge interaction, and 'warm' or personalize the discussion; the tone is friendly, even intimate; includes phatics and emoticons. ['Thanks for your brilliant description of the problems new teachers face—you could have been describing any one of us, I think.']

5A – quotations and paraphrases: 'Every tool carries with it the spirit by which it has been created.'

5B – citations: 'Werner Karl Heisenberg, *Physics and Philosophy*, 1958.'

Figure 1. Fahy et al.'s (2000) Transcript Analysis Tool Coding Categories
Reprinted by permission of the *Alberta Journal of Educational Research*, from Patrick J. Fahy, Gail Crawford, Mohamed Ally, Peter Cookson, Verna Keller, and Frank Prosser, "The Development and Testing of a Tool for Analysis of Computer Mediated Conferencing Transcripts," *Alberta Journal of Educational Research*, Vol. 46, No. 1, 2000, pp. 85–88.

Mapping the TAT Categories to the Squad Categories

Our use of *mapping* in this article refers to the tools being equivalent for measurement purposes. The following section explains how we have mapped the SQUAD within Fahy's (2002) TAT alignments to realize our SQUAD alignments to Garrison, Anderson, and Archer's (2001) framework.

The TAT category 1A includes *vertical questions,* which assumes a "correct" answer exists and that the question can be answered if the appropriate individual is asked or the right source contacted. The TAT category

Table 1. Alignments of Cognitive Presence (Garrison, Anderson, and Archer 2000, 2001) Model With the Transcript Analysis Tool Categories (Fahy 2002)

Alignment	Triggers	Exploration	Integration	Resolution
1	1A, 1B	2A, 4	2B, 5A, 5B	3
2	1A, 1B, 2B	2A	4, 5A, 5B	3
3	1A, 1B, 2B	2A, 4	3	5A, 5B

From "Assessing Critical Thinking Processes in a Computer Conference," by P. J. Fahy, 2002. Used by permission.

1B comprises *horizontal questions*—there may not be one right answer; others are invited to help provide a plausible or alternative answer or to help shed light on the question (Fahy 2002). The SQUAD category Q is a form of words addressed to a person to elicit information or evoke a response. An example of a question within the SQUAD framework is when students seek clarification from the tutor or other students in order to make appropriate decisions relating to the group coursework (Oriogun 2003b). We can, therefore, comfortably infer that the horizontal and vertical questions from the TAT model equate to the definition offered for category Q within the SQUAD framework.

The TAT category 2A includes *non-referential statements,* which contain little self-revelation and usually do not invite response or dialogue; the main intention is to impart facts or information. The speaker may take a matter-of-fact, a didactic, or even a pedantic stance, providing information or correction to an audience that he or she appears to assume is uninformed or in error, but curious and untested or otherwise open to information or correction. This type of statement may contain implicit values or beliefs, but usually these are inferred and are not as explicit as they are in TAT type 3 *reflections* (Fahy 2002). The SQUAD category U is normally not in the list of categories of messages stipulated by the instigator of the task at hand. This tends to happen at the start of the online postings. Students may be unsure of what the message is supposed to convey. In most cases, it falls within one of the four classified categories (Oriogun 2003b). It is, therefore, reasonable to infer that the U category within the SQUAD framework has a direct mapping with the 2A category within the TAT model.

The TAT category 2B *referential statements* comprises direct answers to questions or comments that refer to specific preceding statements (Fahy 2002). The SQUAD category A is a reply, either spoken or written, as to a question, request, letter, or article. Students are expected to respond to this

Table 2. **The SQUAD Approach: Cognitive Indicators Coding Categories Descriptors (Oriogun 2003b)**

Message Category	Description	Example	Cognitive Indicators
S Suggestion	The process whereby the mere presentation of an idea to a receptive individual leads to the acceptance of the idea.	Students engage with other students within their coursework groups by offering advice, a viewpoint, or an alternative viewpoint to a current one.	Elementary classification In-depth classification Inferencing Judgment Application of strategies
Q Question	A form of word address to a person to elicit information or evoke a response.	Students may seek clarification from the tutor or other students to make appropriate decisions relating to the group coursework.	Elementary classification In-depth classification
U Unclassified	Not in the list of categories of messages stipulated by the instigator of the task at hand.	This tends to happen at the start of the online postings. Students may be unsure of what the message is supposed to convey. In most cases, it falls within one of the four classified categories.	Elementary classification
A Answer	Reply, either spoken or written, as to a question, request, letter, or article.	Students are expected to respond to this type of message with a range of possible solutions/ alternatives.	Elementary classification In-depth classification Inferencing Judgment
D Delivery	The act of distribution of goods, mail, and so on.	Students are expected to produce a piece of software at the end of the semester. They all have to participate in delivering aspects of the artifacts making up the software.	Elementary classification In-depth classification Inferencing Judgment Application of strategies

Reprinted by permission from "Towards Understanding Online Learning Levels of Engagement Using the SQUAD Approach to CMC Discourse," by P. K. Oriogun, *Australian Journal of Educational Technology*, Vol. 19, No. 3, 2003, pp. 371–387. Available online at http://www.ascilite.org.au/ajet/ajet19/oriogun.html type of message with a range of possible solutions/alternatives. Also, the SQUAD category S is the process whereby the mere presentation of an idea to a receptive individual leads to the acceptance of the idea, and students engage with other students within their coursework groups by offering advice, a viewpoint, or an alternative viewpoint to a current one (Oriogun 2003b). It is reasonable to accept that the SQUAD categories A and S equate to the TAT category 2B.

The TAT category 3, *reflections*, shows the speaker expressing thoughts, judgments, opinion, or information that are personal and are usually guarded or private. The speaker may also reveal personal values, beliefs, doubts, convictions, and ideas acknowledged as personal. The listener/reader receives both information about some aspect of the world (in the form of opinions) and insights into the speaker. Listeners are assumed to be interested in and empathic toward these personal revelations and are expected to respond with understanding and acceptance. The speaker implicitly welcomes questions (even personal ones), as well as self-revelations in turn, and other supportive responses (Fahy 2002). The SQUAD category S described earlier is focused on what the group has to deliver for their group coursework and does not necessarily deal with significant personal revelation with reference to the TAT definition. However, an individual's personal thoughts on the group's coursework deliverables is part of what is dealt with here.

The SQUAD S category also encourages what is described within the TAT model category 4, *scaffolding/engaging*. Students are expected to initiate, continue, or acknowledge interpersonal interaction, and/or "warm" and personalize the discussion. They do this by agreeing with, thanking, or otherwise recognizing someone else and encouraging or recognizing the helpfulness, ideas and comments, capabilities, and experience of others. The SQUAD category D is the act of distribution of goods, mail, and other items. This is where students are expected to produce a piece of software at the end of the semester. They all have to participate in delivering aspects of the artifacts making up the software (Oriogun 2003b). At this point, students may show their appreciation to part of the group coursework deliverable by responding with comments with real substantive meaning (phatic communion, elevator/weather talk, salutation/greetings, and closings/signatures), and devices such as obvious rhetorical questions and emoticons (Fahy 2002).

The TAT category 5A and 5B deals with *quotations/citations*. This relates to quotations or fairly direct paraphrases of sources and citations or attributions of quotations or paraphrases. Within the SQUAD framework,

Table 3. Proposed Alignment of Cognitive Presence (Garrison, Anderson, and Archer 2000, 2001) in Oriogun's SQUAD Approach by Adopting the Transcript Analysis Tool Model (Fahy 2002) Coding Categories

Alignment	Triggers	Exploration	Integration	Resolution
1	Q	U, S	A, S	S, D
2	Q, A	U	S, D	S, D
3	Q, A	U, S	S, D	S

Note: SQUAD = Suggestion, Question, Unclassified, Answer, Delivery.

category S deals with quotations/citations in exactly the same way as in the TAT model. Table 3 shows our proposed alignments of cognitive presence (Garrison, Anderson, and Archer 2000, 2001) in Oriogun's (2003b) SQUAD approach by adopting the TAT model (Fahy 2002) coding categories based on the TAT mapping articulated earlier. Please note that the SQUAD alignments with TAT are such that, for each alignment, it is possible to have more than one of the categories of SQUAD within the four phases of the practical inquiry model we are considering for this article. Table 3 is our proposed alignment of the cognitive presence (Garrison, Anderson, and Archer 2000, 2001) model with the SQUAD framework by adopting Fahy's (2002) TAT model coding template.

Method

A second version of a tool supporting the SQUAD approach has now been developed: SQUAD v 2.0 (Oriogun and Ramsay 2005). In this article, we report on a pilot study that was conducted to investigate the application of the TAT alignment to the SQUAD approach with the practical inquiry (Garrison, Anderson, and Archer 2001) models. The purpose of this undertaking was to develop a framework capable of describing group-level cognitive engagement. The first study corpus used was the transcript of two groups of software engineering students in a master's program in computing in the first semester of 2004–2005. By the end of the study, in week 12, the first group had posted a total of 725 messages, and the second group had posted a total of 143 messages. The second study corpus consisted of five part-time evening master's computing students. During the second semester of 2004–2005, they posted a total of 171 messages during the first twelve weeks of the study. The three case studies over the year and their contributions to SQUAD message categories are shown in Table 4. A total of 1,039 messages were posted throughout the academic year. Table 5

Table 4. Total Number of SQUAD Postings by Master's Computing Students (2004–2005 Academic Semesters)

Case Study	S	Q	U	A	D	Total
1	132	105	243	157	88	725
2	21	14	66	10	32	143
3	55	18	27	26	45	171

Note: SQUAD = Suggestion, Question, Unclassified, Answer, Delivery.

Table 5. Case Study 1 Results Applying Transcript Analysis Tool Alignment to the SQUAD Approach Using the Practical Inquiry Model

Phases of Practical Inquiry Model	SQUAD No. 1	SQUAD No. 2	SQUAD No. 3
Triggers	14.5	36.1	36.1
Exploration	51.7	33.5	51.7
Integration	39.9	30.3	30.3
Resolution	30.3	30.3	18.2

Note: All table values are percentages. SQUAD = Suggestion, Question, Unclassified, Answer, Delivery.

shows the results of applying the TAT alignment to the SQUAD approach with the phases of the practical inquiry model for Case Study 1; Table 6 shows the results of the same for Case Study 2.

Since the recommendation of his TAT alignments (Fahy 2002), Fahy (2005) has published detailed results in a study consisting of 462 postings, comprising 3,126 sentences containing approximately 54,000 words, generated by a group of thirteen students and an instructor/moderator, engaged in a thirteen-week distance education graduate credit course delivered totally at a distance. We have seized the opportunity to compare Fahy's (2005) findings with our TAT alignment of Oriogun's (2003b) SQUAD approach as described earlier (see Table 3) using the two methods for assessing critical thinking in CMC transcript (Fahy 2005). Table 7 shows our Case Study 1, with students from Group 1. These students posted a total of 725 messages over a period of twelve weeks using the SQUAD approach. Table 8 shows the results for our Case Study 2, Group 2, posting a total of 143 messages over the twelve weeks of the study. Table 9 shows the results for our Case Study 3, Group 3, posting a total of 171 messages over the first twelve weeks of the second semester in 2004–2005.

Table 6. Case Study 2 Results Applying Transcript Analysis Tool Alignment to SQUAD Approach Using the Practical Inquiry Model

Phases of Practical Inquiry Model	SQUAD No. 1	SQUAD No. 2	SQUAD No. 3
Triggers	9.8	16.8	16.8
Exploration	60.8	46.2	60.8
Integration	21.7	37.1	37.1
Resolution	37.1	37.1	14.7

Note: All table values are percentages. SQUAD = Suggestion, Question, Unclassified, Answer, Delivery.

Discussion

When we compare phases of the practical inquiry model with Fahy's (2005) practical inquiry/TAT results and our three case studies' SQUAD TAT alignments (see Tables 7, 8, and 9), we observe more favorable results. Because the SQUAD is a semistructured approach to CMC discourse at the message level, it helps to scaffold students' online learning. There is no need to perform an interrater reliability measure with the SQUAD approach, as the students had to use one of the SQUAD message categories.

In our first case study, with a total of 725 message postings, the SQUAD results applying TAT alignments SQUAD No. 2 shows that the group's overall average contribution to each phase was 32.6% (the average of percentages in Table 7, column 6). This is indeed an ideal result, on the basis that this particular group of students made effective use of all the message categories.

In our second case study, with a total of 143 message postings, the SQUAD results applying the TAT alignments SQUAD No. 2 shows that the group's overall average contribution to each phase was 34.3% (the average of percentages in Table 8, column 6).

In our third case study, with a total of 171 message postings, the SQUAD results applying TAT alignments SQUAD No. 1 shows that the group's overall average contribution to each phase was 41.1% (the average of percentages in Table 9, column 5). Overall, Case Study 3 implies that this group of students contributed, on average, 40.6% postings to each of the phases of the practical inquiry model (the average of percentages in Table 9, columns 5–7). This is indeed a much better result than the results from the first semester of 2004–2005.

One of the reasons the groups of students in our three studies (a total of thirteen in the three groups) made effective use of the SQUAD categories at the message level is that, out of the total marks awarded to the group

Table 7. Comparison of Phases of the Practical Inquiry Model With the Present Fahy (2005) Practical Inquiry/TAT Results and Case Study 1 TAT Alignments

Phases of the Practical Inquiry Model	Practical Inquiry Model Results, Garrison, Anderson, and Archer (2001) Initial Pilot	Practical Inquiry Model Results, Fahy (2005) Present Study	TAT Results, Fahy (2005)	SQUAD Results Applying TAT Alignments SQUAD No. 1	SQUAD Results Applying TAT Alignments SQUAD No. 2	SQUAD Results Applying TAT Alignments SQUAD No. 3
Triggers	12.5	9.4	6.4	14.5	36.1	36.1
Exploration	62.5	74.2	76.4	51.7	33.5	51.7
Integration	18.8	14.6	14.7	39.9	30.3	30.3
Resolution	6.3	1.8	2.5	30.3	30.3	18.2

Note: All table values are percentages. TAT = Transcript Analysis Tool; SQUAD = Suggestion, Question, Unclassified, Answer, Delivery.

Table 8. Comparison of Phases of the Practical Inquiry Model With the Present Fahy (2005) Practical Inquiry/TAT Results and Case Study 2 TAT Alignments

Phases of the Practical Inquiry Model	Practical Inquiry Model Results, Garrison, Anderson, and Archer (2001) Initial Pilot	Practical Inquiry Model Results, Fahy (2005) Present Study	TAT Results, Fahy (2005)	SQUAD Results Applying TAT Alignments SQUAD No. 1	SQUAD Results Applying TAT Alignments SQUAD No. 2	SQUAD Results Applying TAT Alignments SQUAD No. 3
Triggers	12.5	9.4	6.4	9.8	16.8	16.8
Exploration	62.5	74.2	76.4	60.8	46.2	60.8
Integration	18.8	14.6	14.7	21.7	37.1	37.1
Resolution	6.3	1.8	2.5	37.1	37.1	14.7

Note: All table values are percentages. TAT = Transcript Analysis Tool; SQUAD = Suggestion, Question, Unclassified, Answer, Delivery.

Table 9. Comparison of Phases of the Practical Inquiry Model With the Present Fahy (2005) Practical Inquiry/TAT Results and Case Study 3 TAT Alignments

Phases of the Practical Inquiry Model	Practical Inquiry Model Results, Garrison, Anderson, and Archer (2001) Initial Pilot	Practical Inquiry Model Results, Fahy (2005) Present Study	TAT Results, Fahy (2005)	SQUAD Results Applying TAT Alignments SQUAD No. 1	SQUAD Results Applying TAT Alignments SQUAD No. 2	SQUAD Results Applying TAT Alignments SQUAD No. 3
Triggers	12.5	9.4	6.4	10.5	25.7	25.7
Exploration	62.5	74.2	76.4	47.9	15.8	47.9
Integration	18.8	14.6	14.7	47.4	58.5	58.5
Resolution	6.3	1.8	2.5	58.5	58.5	32.2

Note: All table values are percentages. TAT = Transcript Analysis Tool; SQUAD = Suggestion, Question, Unclassified, Answer, Delivery.

coursework for collaborating and negotiating software requirements during the semester, 7.5% of the marks were for using the SQUAD approach (extrinsic motivation). In fact, at the end of the semester the students reported that if no marks had been attached to adopting the SQUAD approach, they would most probably have used other forms of communication, including publicly available online collaborative systems.

Results from a quantitative analysis of the 1,039 total message postings showed that the three groups contributed an average of 32.6% (Case Study 1), 34.3% (Case Study 2) and 41.1% (Case Study 3) of their postings to each phase of the practical inquiry model. On the basis of these and related findings, we conclude that the three groups of students made effective use of all the message categories for cognitive engagement within online groups.

Conclusion

The results from the initial pilot of the practical inquiry model of Garrison, Anderson, and Archer's (2001) study, the practical inquiry results from Fahy's (2005) study, and SQUAD results applying TAT alignments all showed that exploration was clearly the most common type of posting (see Tables 7, 8, and 9). The TAT result and the initial practical inquiry model results showed that the next most common type of posting was integration. This is where the SQUAD approach proved to have shown much better results, in that if one looks at the average posting within each of the phases of the practical inquiry model one sees that, on average, each group contributed approximately the same number of postings to each of the categories. The main reason for this could be that both the critical inquiry model and the SQUAD TAT alignments use the message as a unit of measurement. Furthermore, the SQUAD approach does not require an interrater reliability measure as it is a semistructured method for scaffolding students' learning.

Although we do not have similar concern in this study regarding the category of "other" within the practical inquiry model, this category warrants further investigation. It is worth noting that, in Fahy's (2002) suggested TAT alignments, multiple message categories were not permitted (e.g., in the case of TAT No. 1, the sum total of all the categories is 100% under triggers, exploration, integration, and resolution; see Table 1). However, because of the cognitive indicators governing the SQUAD framework, multiple message categories are permitted (e.g., in the case of SQUAD No. 1, message category S appeared under exploration, integration, and resolution; see Table 3). Perhaps Fahy's (2002) alignments are too restrictive at sentence level. Further testing of the practical inquiry model is required to ascertain its robustness and validity. There is a real need to develop Garrison et al.'s (2001) framework, especially empirically testing it in relation to actual transcripts of online communications.

We believe that through the theorizing and empirical work described herein, we have substantially supported our argument that the cognitive presence realized in this article for the SQUAD approach, using Fahy's (2002) three alignment within Garrison et al.'s (2001) framework together with our three case studies using master's computing students at London Metropolitan University, is a way of empirically validating the cognitive engagement of the SQUAD approach to CMC discourse within groups.

References

Barrows, H. 1996. Problem-based learning in medicine and beyond: A brief overview. In *Bringing problem-based learning to higher education: Theory and practice*, ed. L. Wilkerson and W. Gijselaers, 3–11. San Francisco: Jossey-Bass.

Bridges, E. M. 1992. *Problem-based learning for administrators.* Eugene, OR: ERIC Clearinghouse on Educational Management. ERIC, ED 347617.

Fahy, P. J. 2002. Assessing critical thinking processes in a computer conference. Centre for Distance Education, Athabasca University, Athabasca, Alberta, Canada. Unpublished manuscript. Available online at http://cde.athabascau.ca/softeval/reports/mag4.pdf

———. 2005. Two methods for assessing critical thinking in computer-mediated communications (CMC) transcripts. *International Journal of Instructional Technology and Distance Learning.* Available online at http://www.itdl.org/Journal/Mar_05/article02.htm

Fahy, P. J., G. Crawford, M. Ally, P. Cookson, V. Keller, and F. Prosser. 2000. The development and testing of a tool for analysis of computer mediated conferencing transcripts. *Alberta Journal of Educational Research* 46 (1): 85–88.

Garrison, R., T. Anderson, and W. Archer. 2000. Critical inquiry in a text-based environment: Computer conferencing in higher education. *Internet and Higher Education* 11 (2): 1–14.

———. 2001. Critical thinking, cognitive presence, and computer conferencing in distance education. *American Journal of Distance Education* 15 (1): 7–23.

Hara, N., C. Bonk, and C. Angeli. 2000. Content analysis of online discussion in an applied educational psychology course. *Instructional Science* 28 (2): 115–152.

Henri, F. 1992. Computer conferencing and content analysis. In *Online education: Perspectives on a new environment,* ed. A. R. Kaye, 115–136. New York: Praeger.

Johnson, D. W., and R. T. Johnson. 1996. Cooperation and the use of technology. In *Handbook of research for educational telecommunications and technology,* ed. D. H. Jonassen, 1017–1044. New York: Simon & Schuster.

Oriogun, P. K. 2003a. Content analysis of online inter-rater reliability using the transcript reliability cleaning percentage: A software engineering case study. Paper presented at the ICEIS 2003 conference, April, Angers, France.

———. 2003b. Towards understanding online learning levels of engagement using the SQUAD approach to CMC discourse. *Australian Journal of Educational Technology* 19 (3): 371–387. Available online at http://www.ascilite.org.au/ajet/ajet19/oriogun.html

———. 2005. Introducing a dedicated tool for facilitating a semi-structured approach to CMC messaging, Workshop on e-Learning Online Communities, eLOC 2005, CD-ROM Proceedings of the 3rd ACS/IEEE International Conference on Computer Systems and Applications (AICCSA-05), January, Cairo, Egypt.

Oriogun, P. K., and J. Cook. 2003.Transcript Reliability Cleaning Percentage: An alternative interrater reliability measure of message transcripts in online learning. *American Journal of Distance Education* 17 (4): 221–234.

Oriogun, P. K., F. French, and R. Haynes. 2002. Using the enhanced Problem-Based Learning Grid: Three multimedia case studies. Paper presented at the ASCILITE conference, December, Auckland, New Zealand. Available online at http://www.ascilite.org.au/conferences/auckland02/proceedings/papers/040.pdf

Oriogun, P. K., and E. Ramsay. 2005. Introducing a dedicated prototype application tool for measuring students' online learning levels of engagement in a problem-based learning context. Paper presented at the IASTED, special session, International Conference on Education and Technology (ICET 2005), July, Calgary, Alberta, Canada.

Zhu, E. (1996). Meaning negotiation, knowledge construction, and mentoring in a distance learning course. In *Proceedings of selected research and development presentations at the 18th National Conference* of the Association for Educational Communications and Technology, Indianapolis, IN. ERIC, ED 397849.