Reaching for the moon: a technique for introducing complex methods at Levels 5 and 6

Irene Davey
Business Studies, Information Technology
Bay of Plenty Polytechnic
Tauranga, NZ
irene.davey@boppoly.ac.nz

Don Gotterbarn
Visiting Professor of Software Engineering Ethics
Computer and Information Sciences
Auckland University of Technology,
Auckland, NZ

Many NACCQ's Diploma in Information of Communication Technology Level 5 and 6 students have limited knowledge of software development. This problem is especially evident in their understanding of system development life cycles, analysis of systems, or writing programmes. We report on the results of a project designed to teach students, even at this stage, to apply complex techniques and standards of system development.
To test the effectiveness of this process, a software development review method was adopted and gradually introduced though several lower level courses in the ICT curriculum. This process of structuring the presentation of the material successfully met the objectives of developing student's skill to analyse, document and audit a software project and it nurtured interpersonal skills necessary for successful teamwork. The procedure used supports the introduction of any complex method to students who might not appear to be academically prepared. Although a particular software tool, the SoDIS Project Auditor was used as the illustrative methodology; we have clear indications that any complex process can be introduced using our distributed techniques. This paper describes an actual application of this teaching approach, its outcomes, and suggests other applications of the method.

1. INTRODUCTION

It is often difficult, because of time and skill constraints, to introduce advanced topics at NZQA Level 5 and 6. This paper describes the successful introduction of a particular set of advanced skills and knowledge to this level of preparation. We have two goals in this paper. The first is to describe the particular skill and how it was introduced in the hopes that others can introduce this same set of skills. The second goal is to encourage you to 'shoot for the moon' in your teaching, and we offer this example to encourage you to introduce interesting and difficult concepts of your beginning students. The rewards are worth the extra design effort.

1.1 Situation

Many NZQA Level 5 and 6 students have only limited knowledge of the application of a system development life cycle or application of it to analysing and developing a database or writing a software programme. We report on the results of a project designed to teach students, even at this stage, to apply complex techniques and standards of system development. To test the effectiveness of this process, a complex software development review process has been adopted and gradually introduced though several lower level courses which successfully meet objectives of developing student's skill supportive to being able to analyse, document and audit a software project. It also introduces interpersonal skills necessary for successful team development. Although a particular software tool, the SoDIS Project Auditor (SPA) was used; the method of its introduction across several courses can be used to incorporate other software review tools and processes. This approach makes it possible to introduce complex material at an earlier stage.

The SoDIS (Software Development Impact Statement) process presumes some basic knowledge of systems development, NZQA Level 5 and 6 students have varied backgrounds but most do not present with some of the backgrounds that might be an advantage in using SoDIS. These include: project management, either a very wide use of databases or use of programmes other than Microsoft, and higher management skills (especially in the Information Technology field).

1.2 The broad teaching goal

We believe that good software developers must be able to think outside the square of a specific programming language or a specific application when they develop custom software. Critical thinking is

encouraged by adoption of a review process, which systematically forces them to consider how people other than the customer and the developer (stakeholders) are related to a particular project task. The software development process that facilitates this is the SoDIS (Software Development Impact Statement) and its implementation software - the SoDIS Project Auditor (SPA).

2. REASONS THIS PROCESS (USING SODIS) WAS CHOSEN

The process using SoDIS incorporates a particular interactive learning style. The limited period of contact between staff and students imposes constraints that are not conducive to allowing time for self-discovery ("here is a software package, go play with it and see if you can answer these questions about it"). To overcome this difficulty, we introduce SoDIS Project Auditor to the students in a way that can build upon knowledge that they bring from previous areas of learning (Billett, 1996).

Briefly SPA provides a framework of questions related to software project development. Those questions are provided at several levels. Some questions require a simple 'yes' or 'no' answer to clear fact questions like those is common software risk analysis checklists; e.g. Is there an agreed upon project plan? A negative answer to this question requires that an action item to build a plan be undertaken. The presence of this question serves to educate students that at least some developers consider a plan important. Other questions generated by the SPA require judgement about much broader issues in software development; e.g. 'Might the development of an Internet filter for a classroom negatively impact a particular stakeholder?' This requires them to interact with the question by first determining who the stakeholders are. If they answer 'yes' to this question for any stakeholder, such as the teacher, the SPA requires them to go further and address possible solutions to the potential harm they have identified.

The SPA enables students to work within a cognitive apprenticeship using constructivist methods acquiring expertise within its boundaries (Farmer, Buckmaster, & LeGrand, 1992). Research on the use of the SoDIS has found that it works best in a group decision-making mode. It questions are addressed individually, then by two or more students. The analysis and recording of answers requires that they work together and find ways to come to agreement about identified issues.

Use of SoDIS process supports academic introduction of the topics of life cycle phases, stakeholders, ethics, and tasks. Furthermore, academically SoDIS sustains these topics integration with concepts of individual and interpersonal work methods of systems thinking, impacts on users, user input and participation, critical thinking and the completion of the cycle by reflection on needs and performance.

SoDIS Project Auditor teaches students interactively at all levels the (importance of):

- Significance of developing phases – specifically, feasibility, requirements analysis and detailed analysis - supporting the documentation and auditing what is and has been included and completed within the life cycle. (SPA asks different questions depending on the development phase being analysed)
- Identification of stakeholders – those who are directly impacted by the project immediately (primary) as well as those that are extended stakeholders (secondary). (SPA provides list of preliminary stakeholder roles for each type of project. Specific stakeholders of a role type are identified by the students. This forces them to think outside the square.)
- Integration and supporting identification of tasks, and relating them to stakeholders and issues already identified.
- Ethics – supporting moral issues and concerns as well as financial needs within project planning. (The questions SPA generates were derived from international software codes of practice and codes of ethics.)

These areas are considered within SoDIS and are supported by the use of templates for a particular business area. SoDIS supports the consideration of general items that are identified in the templates but still allows additions to support inclusiveness of individual differences of each project. Student can add new questions, stakeholder types to any analysis.

The difficulties students' first encounter doing a SoDIS analysis in teams makes them receptive to

techniques that facilitate teamwork. SoDIS is academically supportive of the introduction to concepts of work skills: teamwork, systems thinking, impact on user interaction, user input and participation, critical thinking and reflection. Teamwork may be supported in the preparation to use SoDIS building concepts of individually reflecting on the project in hand, analysing considerations in pairs to broaden areas of contemplation and by brainstorming as a team ensure the widest areas are put forward for deliberation, critically thinking about the impacts and participation of users within the system life cycle. It is interesting that these advantages come from the SoDIS process and do not require the initial use of a distracting piece of software, as we shall see below. Professional concepts introduced and supported by SoDIS include:

- Systems thinking can be supported within the structured concepts of SoDIS
- Understanding impact of requirements or development tasks by asking questions and supporting feedback by use of the structure supported in SoDIS.
- Critical thinking is implemented by the consideration that is required after all environmental impacts tasks and the possible stakeholders and actions are identified within the first stage of SoDIS's system development life cycle – the feasibility stage. This is especially supported when the questions are formed to identify if there is any impact on a stakeholder.
- Reflection on needs and performance is supported and required at all levels but at the Requirements and Detailed analysis stages of the system development life cycle these skills are endorsed and all possible impacts must be addressed or at the very least acknowledged that a specific item has not been addressed.

3 HOW SoDIS CAN BE USED WITH BEGINNING STUDENTS

Beginning students (in for example NZQA Level 5 and 6) have varied backgrounds but most do not present with skills supportive to being able to analyse, document and audit a project. Some of the backgrounds that might be an advantage in using SoDIS besides project management are; either a very wide use of databases or use of programmes other than Microsoft, higher management skills (especially in the Information Technology field). At this stage the student still may have limited knowledge of the application of a system development life cycle or application of it within analysing and developing a database or writing a programme. Even if they know the theory they find it hard to accept it as a real or accepted methodology.

Within this range of students the allocation of administrative restrictions on time and the integration of the sections in the system development life cycle are not as comprehensive and optimally supportive as an academic might wish. For example with time, where the students used to have 48 hours out of a recommended 70 hours of contact for each course they are now allocated 32 hours. This appears to be compounded by the move towards running the courses over a semester instead of by the term so that the students are involved with nine individual courses, each having two hours contact a week. Besides impacting the length of time between each course's weekly class, the built in spiral, that appeared to be inherent when only four or five courses were taught a term, has largely been lost.

Level 5 students, with care, can have the concepts of SoDIS introduced and built upon within and over a range of courses, from courses like the NACCQ range of Diplomas and Information Technology based degrees. For example, within the Diploma's first year, students are introduced to several areas of study, which may be integrated into the concepts within SoDIS - systems overview (SO500) for the system development life cycle, ethical principles in ET600 programming principles and development in PP500 and PD500, and databases in DB500. All except DB500 are compulsory papers.

At Level 5, students are better served by being introduced to the concepts of SoDIS and not using the SPA itself until they have reached Level 6 because of the lack of depth of prior skills and restricted time. This is critical to the process. Students are not distracted by the fine details of a piece of software but get to approach the process from a conceptual level. A simple case study is used which also does not require that they have or use any programming skills. The concepts that have been identified as the initial steps of understanding in the use of the SoDIS and the SPA programme are:

- Identification of stakeholders – immediate and extended.
- Identification of tasks that are assigned to immediate and extended stakeholders.
- The identification of any ethical concerns and issues that are or may be attached to the tasks and stakeholders.
- Record the details and solutions to help modify the list of development task.

To support the introduction of SoDIS to Level 5.0 students a class strategy was designed and developed using a presentation introducing a simple handout called 'Race into SoDIS Methodology'.

3.1 The class room process

The process described here requires four one-hour class contact sessions and approximately the same amount of time out of class work and reflection.

The first session is used to introduce SoDIS using the PowerPoint and work through a class/student suggested project as a class brainstormed, on the whiteboard and worksheets, mini "SoDIS" project. Once the mini project is selected out of those proposed by the class it is quickly worked through (not in depth) to show how the concepts and process might be approached. The steps are:

- Identify Immediate and Extended Stakeholders.
- Identify tasks or work breakdown packages.
- Record potential issues – Questions for every stakeholder, related to every task.
- Record the details and solutions to help modify the development plan – as an Action Item.

The students are then given a case study to take home and are asked to fill in the Stakeholder form (solo), with all stakeholders that they can think of that have connection with the case. One recent case was: 'You have been asked by a local primary school teacher to develop and install an Internet filter on her classroom computer which limits access from the classroom computer only to web sites which she approves.'

At the second class the students' pair up and compare lists analysing and reflecting on choices and supporting arguments for each being part of the list. Then the class brings all the lists together each pair in turn adding to a whiteboard list until all stakeholders are listed – then the class discusses, analysing, supporting and refuting each of them and identifying if they are immediate of extended stakeholders. They immediately recognize students and the teacher as stakeholders. Their discussion discloses other stakeholders. The system development life cycle steps (learnt in SO500) is given life by giving a task sheet to students to fill in the all the tasks that might need to be included in the project using the life cycle framework and to think of what and how the tasks relate to each of the stakeholders. As homework they are given the sheet that requires them to relate task and stakeholders with ethical issues.

The third class follows the same active learning jigsaw format as the first class – working in pairs then as a class within a teamwork frame analysing the tasks and relating the task and stakeholders with ethical issues. Students are asked to complete this sheet in pairs and commence on putting the last sheet, the Ethical issues - concerns into audit questions for homework. Notice that so far this does not require advanced knowledge of computing but merely a simple user understanding of the Internet.

In the last class two sets of pairs analyse and reflect on the project sheets paying special interest in the way that the questions are posed. These questions could be phased where a 'yes' answer identifies the existence of a problem solved or they could be phrased so that a 'no' answer indicates the absence of "something" (required by good systems development) requiring some action to fix. It requires a lot of extra thought and reflection to phrase questions in the manner that the questions are answered in the negative if action is needed. If also requires them to think about the relevance of the life cycle to this simple computing task.

Students are then required to work in pairs to commence a case study's set of sheets then with another pair to complete analysis of the homework. Any changes or updates are shown typed in italics then the work is handed in. This is not expected to be a definitive set of all stakeholders, tasks and issues, but must show the students ability to analyse and critically think and reflect on each set of sheets.

Students become aware of the improved quality of their analysis by working in pairs and groups. The collaborations appeared to be a major component

in the success when it followed through supporting individuals initial working and thinking "it" through phase. (Bibby, 1998)

4 SUMMARY AND CONCLUSION

Although they have not used the SPA software they have applied a SoDIS like process and seem the relevance of the software standards to achieving good results. It has made a clear difference in their experience in other courses. As new elements were introduced into their case, such as, the computers are networked meaning any restriction on computer's Internet accessibility would impact the entire school - students immediately see the impacts of this on the design and development of the system. Level 6.0 teaching colleagues have reported positive spin-offs from the introduction to SoDIS in the Level 5.0 curriculum. The courses that tutors have given positive feedback without solicitation include Project Management (PJ600), Systems Analysis (SA600), Programming (varying), Database (DB600), Software packages (SP700) and User Support (US500 and US600).

Within overall feedback, consensus was that the students now had a better understanding of how the system development life cycle might be used (based on the SoDIS steps); forming a foundation for systems thinking constructing a scenario, case study or project even when commencing at different stages (as each of the identified courses support); and the students appeared to be better positioned for critical thinking and reflective analysis.

User Support found that the analysis and expectation of user interaction appeared to be enhanced and wider in scope than previously. SP700 and PJ600 courses reported that students understood and could implement and integrate the system development life cycle within their projects – giving far greater thought to the tasks, ethics and possible impact on stakeholders than previous class groups. As one student said "I can see why we need so much structure and documentation now, it allows us to track what we are up to and what we still have to do". PJ600 found that the students were able to use the concepts taught in the "Race into SoDIS" especially the collaboration in the early stages. This was found to assist greatly in having a maturity of breadth and depth to the initial analysis and allowed far grater professionalism than is normal at this level of teaching. Quantifying the feedback and other aspects is planned for the next cycle.

REFERENCES

Bibby, M. (2003). Code of ethics. AARA. Retrieved from the WWW 10 February 2003, <http://www.aare.edu.au/ethics/ethcfull.htm>

Billett, S. (1996). Towards a model of workplace learning: The learning curriculum. Studies in Continuing Education 18, 1: 43-58

Davey, I. J. (2003). Race into SoDIS methodology. (PowerPoint document). Tauranga, New Zealand: Bay of Plenty Polytechnic

Farmer, J. A., Jr., Buckmaster, A., and LeGrand, B. (1992). Cognitive apprenticeship. New Directions In Adult And Continuing Education, no. 55: 41-49. (EJ 456 732)

Gleason, D. (2001) A software development solution. Retrieved on 7 August 2003 from <http://www.itqual.com/docs/A-PM-Solution.pdf>

Gotterbarn, D. (1990). Computer ethics: responsibility regained. Address to the Computers and Quality of Life Conference, Washington DC, September 16

Gotterbarn, D. (1999). Cleanroom, PSP, and the Software Development Impact Statement: Developing the Right Attitude. 12th Conference on Software Engineering Education and Training. Retrieved on 7 August 2003 from <http://csdl.computer.org/comp/proceedings/cseet/1999/0131/00/01310080abs.htm>

The National Qualifications Framework: About the Framework. (nd) Retrieved on 7 August 2003 from <http://www.nzqa.govt.nz/framework/rol/docs/leveldes.doc>

Weckert, J. & Adeney, D. (1997). Computer and information ethics. Westport, CT: Greenwood.