Leveraging Student Projects into Case Studies for Educational Purposes

Joseph Kasser DSc, CEng, CM
University of South Australia
Mawson Lakes Campus
Mawson Lakes, SA 5095
joseph.kasser@unisa.edu.au

Abstract Instructors teaching postgraduate subjects in systems and software engineering tend to have difficulty finding case studies that cover much if not all of the life cycle. This paper identifies the need in several systems and software-engineering subjects and then describes how student projects in software engineering in one class can be designed for reuse as case studies in other classes.

The paper is based on the need as noted in the software engineering subjects taught in the Graduate School of Management and Technology at University of Maryland University College (UMUC). The paper describes how and why the student projects were defined for the software engineering project class (MSWE617) so as to be readily convertible into Case Studies for other classes in the program as well as for the subjects at the University of South Australia.

Keywords Systems engineering, software engineering, engineering education, collaborative learning.

1 INTRODUCTION[1]

Students in postgraduate classes in systems and software-engineering invariably request samples of documents and presentations that need to be produced at various points in the systems and software development life cycle (SDLC). However, such samples tend to be difficult to obtain from industry. Moreover, if obtained, the students tend to be overwhelmed by the details of unfamiliar projects the documents describe, and get bogged down in the details.

Various product standards that are available provide templates for the contents of documents in the form of 'tables of contents' and 'check lists'. However, they fail to provide any examples of good and bad contents.

The requirement is thus to provide sample products from small projects that can be comprehended in a short period of time, so that the students can focus on template for the content, instead of the details contained within the samples.

Looking at the problem from a system's perspective, the students produce sample documents in the classes and there

[1] This work was partially funded by the DSTO Centre of Expertise Contract

are several classes. Some instructors provide students in a current iteration of a class with a sample produced by students in previous iterations. It should be possible extend this concept to both institutionalize it, and use documents produced by students in a class in one subject for students in class in a second subject[2].

A second need is for a realistic background scenario for student projects. While the instructor can provide some information about a hypothetical system, many details are lacking, so that the richness of the project cannot easily be obtained in a 13 to 15-week semester. The students have to go through the team building process, learn the "background" to the project, and produce a product within a single semester.

Another need is to obtain up-to-date case studies of different projects to examine how and why they succeeded or failed so as to provide "lessons learned" information. Such information tends to be scarce.

2 REUSING STUDENT PRODUCED DOCUMENTS

Student produced documents have been reused in the manners described in this below to meet the needs described above.

2.1 The Student Enrollment and Course Tracking System (SECTS) Project

The SECTS project (Kasser and Williams 1999) was introduced in late 1997 to meet the needs of postgraduate students at University of Maryland University College (UMUC) who were enrolled in the Master of Science degree in Computer Systems Management (CSMN). As part of the requirements for the degree, students specialized in one of four tracks or majors. One of these tracks was the Software Development Management (SDM) track. Students in this track took classes in 'system and software requirements' (CSMN 645), 'design' (CSMN 646), 'testing' (CSMN 647), and 'maintenance' (CSMN 648). It was difficult for the students to get the full benefit of working on projects because

- There was little connectivity between the courses in the track
- While the advisors recommend the classes be taken in sequence there was no requirement to do so, and consequently most of the students did not take the classes in the optimal sequence.

The purpose of the SECTS project was to

- Provide a series of linked projects that would allow the students in a class in one subject to become familiar with the background to a specific project in a minimal amount of time.
- Provide some continuity between the different classes in the SDM track of the CSMN program.
- Minimize the effect of lacking prerequisite knowledge in students taking the classes out of sequence.
- Allow each of the students in each class to perform the project without depending on information from the then current iteration of the other classes in the other subjects (minimized coordination problems).

The first problem was to find a system the project could be based on so that the students would become familiar with it

[2] With appropriate protection of author's identity

in a very short period of time. The Interactive Registration and Information System (IRIS) at UMUC provided such a system The students were already using it to register for classes, check out grades, and add and drop classes. They thus had a feel (undocumented knowledge) for the way the system did and should behave.

The SECTS Project was designed with this undocumented knowledge in mind, and examines the different aspects of the SECTS during the various parts of the Software Development Life Cycle (SDLC). The SECTS Project scenario was as follows.

Hypothetical University (HU) not UMUC was experiencing problems with the SECTS used to enroll students and track their courses. The focus of each class was as follows:

- System and Software Requirements (CSMN 645) – developed the system and software requirements for the SECTS.
- Software Development and CASE Tools (CSMN 646) - designed the SECTS based on the requirements provided by a combination of earlier iterations of CSMN 645.
- Software Verification and Validation (V&V) (CSMN 647) - verified the requirements for the SECTS provided by a combination of earlier iterations of CSMN 645 and developed a Test Plan for validating the design.
- Software Maintenance (CSMN 648) - examined the SECTS from the maintenance and change control perspective using documents produced by the other classes within the context of a scenario introduced by the instructor.

The combination of documents generated by different iterations of classes helped provide anonymity to the authors and introduce "errors". The SECTS project made an excellent framework for classroom discussion and collaborative learning. It provided a project for each of the classes in the SDM track. The students in each class were split into teams. Using collaborative learning techniques, each team worked independently to produce a product based around common guidance, lectures, discussions, and reviews. The student teams, having different skill sets and experience mixes tended to produce different products. Consequently, part of the project reviews in each class covered the differences between the approaches used by the teams (Meta-Project). This tended to bring the effects of the real world into the classroom. In addition, some students were enrolled in two subjects at the same time, and were able to leverage knowledge gained in one to produce a better product in the other.

The students thus experienced the entire SDLC of a system in up to four semesters. They learned by collaborating to apply knowledge gained in the subject, previous education, and experience, to their assignments in the phase of the SDLC covered by the class. They did this handicapped by the consequences of poor performance (poorly produced documents) in a previous phase. Consequently, they developed an awareness of the effect of poor performance in any phase of the SDLC on its later phases.

This concept mimicked the real world wherein they make decisions based on incomplete and inadequate information. Several students complained (within their groups and during class sessions) that they were not provided with adequate information to get started on the project. These students did not want to

have to figure out how to get the rest of the information that was needed in order to complete a project (i.e., Software Maintenance Plan) on their own, they wanted it handed to them. They did not seem to realize they were also being taught that to succeed in the real world of the SDLC, they needed to take a more proactive approach in getting the required information from the customer. The instructor had to reiterate the three stages of a project and how they differ between industry and academia:

- **Traditional classroom** – confusion, clarification, and completion.
- **Real word** - confusion, completion, and clarification (perhaps).

It seemed that in general, and based on a small sample, that the more work experience a student has had, the greater ability the student has to deal with vagueness. This is a critical skill that did not seem to be taught anywhere else in the program. The cross-fertilization of knowledge and team interaction helped the students in acquiring this critical skill.

The SECTS project worked so well that a second project was initiated about 18 months later. It dealt with the need for HU to open a web based distance-education learning environment.

2.2 Student Provided Case Studies

Many of the students in the CSMN program were mature with years of experience[3]. They provided insight and additional knowledge in the projects. In addition, in my iterations of CSMN 647 and CSMN 648 the students were offered the opportunity to write and present term papers describing their experiences in projects that were in trouble. The papers adhered to the following instructions:

- Document a Case Study. Students had to write a scenario for the paper based on personal experience.
- Analyze the scenario.
- Document the reasons the project succeeded or ran into trouble.
- List and comment on the lessons learned from the analysis.
- Identify a better way with 20/20 hindsight.
- List a **number of situational indicators** that can be used to **identify a project in trouble** or a successful project **while the project is in progress**.

The identifications were removed from their presentations, and the cases were reused in subsequent iterations of the subject. In the web- based classes, by the year 2000, the students were producing audio enhanced PowerPoint presentations using streaming technology. These presentations remained on the class web site for viewing as sample in further iterations of the subject. In addition

- After two iterations of CSMN 647, 19 students had produced papers that identified 34 different indicators for projects in trouble. Their findings provided a baseline for further research that resulted in a published conference paper (Kasser and Williams 1998) that has become required reading in several subjects at

[3] These students were employed in the workforce and earned their degree by studying part time, mostly in the evenings. Their employment positions ranged from programmers to project managers. Some also had up to 20 years of experience in their respective fields.

UMUC and the University of South Australia.

- At least one student upgraded her presentation and presented it at an international conference (Garlow 2000).

3 THE SOFTWARE PROJECT CLASS (MSWE617)

This subject is the capstone class in the Master of Software Engineering program at UMUC. It may be considered as a comprehensive examination covering the application of the tools, skills and techniques the students have acquired in the course of their studies. This subject provides experience in applying software-engineering techniques by giving the students an opportunity to produce software working in teams under the schedule constraints commonly experienced in industry. The instructor is supposed to emulate the vagueness shown by typical customers in describing requirements. The instructor serves as a guide and mentor, not as a traditional teacher. The students are expected to have acquired the knowledge of what to do and how to do it from the prerequisite subjects. It is up to the students to form their own teams (organization) and schedule their work to meet the deadlines imposed by the contract (syllabus). The students are given the chance to join a team to work on a project based on factors that include

- Their level of interest in the topic.
- Other students that they wish to work with, or do not wish to work with.

The projects were picked so that they were small enough to be completed in a single semester, yet complex enough to require a team to produce the product and pertinent documentation. In addition, students were able to propose projects that benefited their employers. These projects were scoped and accepted on the condition that they were able to attract other students to work on the project and agreed to get no outside help in the implementation of the project. Typical instructor determined projects were

- The Webforum, a web-based asynchronous conference (Kasser 2000). This project ran twice. The first class wrote a program in Lotus Domino which had the capability to interface to UMUC's WebTycho Web-based learning environment. The second class produced a Perl script based design that is on-line for use as a demonstration site at http://www.seecforum.unisa.edu.au/webforum/
- A Desk Copy textbook ordering and tracking system
- A multi-user Farleyfile to allow faculty to document interactions with students in an on-line shared database.

Each project was also designed to be of some benefit to the institution, provide samples for other subject as well as test the student's knowledge of the SDLC and its implementation. The students had to cycle through the

- Operations Concept Review.
- Systems Requirements Review.
- Preliminary Design Review.
- Critical Design Review.
- Delivery Readiness Review.

The class did not meet regularly, only at the formal design reviews. Students were free to meet in between times, as and when they decided to do so. There was one exception, in Semester 1 of 2000 one group in the web assisted class had to complete the entire project

on-line without face to face meetings and succeeded (Kasser 2001).

The teams in MSWE 617 had to produce the following products

- Review packages
- Project management plan
- Software requirements document
- Test plan
- Software design document in presentation format
- Test procedures
- Software programmer's manual
- Application user manual
- Source code
- Test reports
- Management reports
- Installation instructions
- Working application software.

Delivery of documents was in two phases. The first deliverable was a draft document. The later in the semester the draft was delivered, the more detailed the draft document was expected to be. Draft documents were expected to contain an annotated outline showing what would be in the final version of the document. Draft documents were due at their respective reviews; final versions were due as specified in the syllabus. The phased delivery method was employed so those students who had to produce documents associated with the earlier reviews were not unduly penalized. The set of delivered documents was then made available for use in other classes. Each review package and instructor's comments on same, were left on the class web site and are available for study by students in other CSMN and MSWE classes.

4 ISSUES REGARDING INTELLECTUAL PROPERTY

Reuse of material raises the issues of copyright and ownership of intellectual property. In UMUC, ownership of student produced material is accorded to the students. Thus in the scenarios described in this paper, the students were notified ahead of time of the intent to reuse their material after suitable editing. These are serious issues which must be addressed by the instructor in accordance with the institution's policies on the topics before any reuse is made.

5 CONCLUSIONS

Designing student projects to produce products suitable for use in other Information Technology classes is not difficult. It just requires the appropriate outlook.

6 REFERENCES

1. Kasser, J E, Williams, V R, 1999, The Student Enrollment and Course Tracking System Meta-Project, PICMET, Portland,

2. Kasser, J E, Williams, V R, 1998, What Do You Mean You Can't Tell Me If My Project Is in Trouble?, First European Conference on Software Metrics (FESMA 98), Antwerp, Belgium

3. Garlow, T, 2000, The Untimely Death of Air National Guard Management Information System (ANGMIS), The Systems Engineering Test and Evaluation Conference (SETE 2000), Brisbane, Australia.

4. Kasser, J E 2000, The Web-Conference: A Case Study, The INCOSE - Mid-Atlantic Regional Conference, Reston, VA.

5. Kasser, J E 2001, Using Systems Engineering to Create and Guide a Successful Student Software Engineering Project Class when the Instructor is Half-Way around the World, Australasian Conference of the Engineering Management Educators, Wollongong, July 2001