The Evolution of Student Systems Development Projects in the Post-graduate Honours Degree Programme

Lydia Palmer
Rhodes University, Grahamstown, South Africa

l.palmer@ru.ac.za

Abstract

The Department of Information Systems at Rhodes University includes in its post-graduate honours degree programme a systems development project that requires groups of students to develop an Information System from initial requirements elicitation and systems analysis to post-implementation review. The Project not only provides students with a near-real world experience, but also draws together, in one exercise, the curriculum into a holistic package.

This paper briefly investigates the recommended curriculum for Information Systems students and notes the inclusion of a Systems Development Team Project in the curriculum. It then tracks the evolution of the Systems Development Project in this Department over a number of years and presents details of the current status of the Project.

Keywords: Curriculum, Development, Information Systems, Systems Development Project, Team, Methodology

Introduction

Systems development projects are incorporated into most Information Systems (IS) curricula. The Department of Information Systems at Rhodes University is no exception. At Rhodes University, systems development projects are incorporated into the curriculum in second, third and honours level.

Systems Development and curriculum recommendations as proposed by the IS 2002 Model Curriculum are investigated and the general curricula as recommended by bodies such as the Association for Computing Machinery (ACM), the Association for Information Systems (AIS), and the Association of Information Technology Professionals (AITP) are discussed. It is noted that the inclusion of a Systems Development project into the Information Systems (IS) curriculum is not uncommon. Many other institutions have successfully implemented team projects into their curriculum, and reasons are cited for the relevance of this practice.

This paper discusses the manner in which the projects have been implemented at the Honours level over the last 26 years. As the department has evolved, the nature of the projects has changed. Three generations of the implementation of systems development projects are discussed. Of particular importance are the criteria that should be met during the implementation of an information systems project.

Background

Rhodes University is a public higher education institution with an annual enrolment of approximately 6000 students. The Department of Information Systems is housed in the Commerce Faculty. Current enrolment in the department is approximately 360 students of which 23 are registered for IS Honours or the fourth year of study.

The Department was established in 1982 as a sub-department of the Department of Management later becoming an independent Department of Information Systems in the Commerce Faculty in 1985.

Two main themes of the Department at both undergraduate and postgraduate levels are Systems Development and IS Management. Since the IS Honours programme was introduced in 1991, the curriculum has included a systems development project as one of the core components. Currently the Systems Development Project comprises 20% of the overall mark for the programme.

Systems Development

According to Avison and Fitzgerald (2006, p. 23), an "Information System is a system which assembles, stores, processes and delivers information relevant to an organisation (or to society), in such a way that information is accessible and useful to those who wish to use it, including managers, staff, clients and citizens. An Information System is a human activity (social) system which may or may not involve the use of computer systems".

Information Systems Development is the way in which Information Systems are conceived, analysed, designed, and implemented. These activities of Information Systems Development are mostly included in the curriculum for Information Systems courses (Avison and Fitzgerald, 2006, p. 23). It is, however, important to afford students the opportunity to put these activities to practice.

Aspects such as teamwork, testing, maintenance and system integration should also be considered when discussing Information Systems development. The question has always arisen as to how these should be incorporated into the curriculum and how should they be practiced.

Systems Development Project Curriculum Recommendations

The ACM, AIS and AITP have come together on more than one occasion to put together a curriculum for Information Systems.

The latest curriculum produced by these organisations is the IS 2002 Model Curriculum and Guidelines for Undergraduate Degree programmes in Information Systems which mentions that systems development projects should be included in the curriculum. (Gorgone, Davis, Valacich, Topi, Feinstein, & Longenecker, 2002).

Examples of these are shown below:

IS 2002.7 - Analysis and Logical Design, make mention of a project in their discussion section, which states that a "project of limited scope will be designed during this course." Further in the same discussion, mention is made of project management being taught and used to control "the team project" and that scheduling and completing individual and group actions will be used to ensure project milestone completion.

IS 2002.8 – Physical Design and Implementation, mention of a team oriented project environment to design and implement a departmental Information System requiring an enterprise level database is made in the discussion section of the module description.

IS 2002.9 – Physical Design and Implementation in Emerging Environments *(Prerequisites: IS 2002.2 and IS 2002.8)* is the module where "students are expected, as their main activity to implement an Information System in an emerging systems environment." It is clearly stated that "Teams will use project management principles to implement an Information System." The discussion for this module states that "students will utilize a contemporary development environment to implement a project that spans the scope of the previous courses." The discussion also states that "Implementation standards should be developed by the students and used rigorously as Project teams complete a significant system. A conversion and training plan should be developed and implemented involving hardware, data, people, and software systems".

In the specifications for **IS 2002.10 – Project Management and Practice**, it is stipulated in the catalogue, that advanced majors will engage in and complete the design and implementation of a *significant Information System*.

Although there is no module description dedicated to a Systems Development project – it seems clear that the intention of the Model Curriculum and Guidelines is for a project to be carried out at some stage during the course.

Systems Development Projects

The inclusion of systems development projects in IS curricula is not new. However, experiences of other institutions in the implementation of these projects are worth discussing.

According to Torngren, Grimheded, & Adamsson (2007), their experience of running projects in an iterative manner created a new dimension to an already successful process. They concluded that project-oriented courses are usually presented at the end of the course, which act as an examination process where students can practice and reveal their knowledge acquired during earlier years of study.

Smarkusky and Smith (2004) state that the incorporation of team projects throughout the curriculum, promotes the understanding of an organisational structure, a team project experience and the impact of information technology, which is lacking in students who have graduated and not participated in any systems development team projects.

Anewalt, Polack-Wahl, Beidler, and Smarkususky (2003) state that group projects are vital for a complete education. They state that many professors consider group projects to be a waste of time, and difficult to implement. These encourage academic staff to take the trouble to introduce group projects, as these do add that vital component to education.

Sherrell and Shiva (2005) discuss the need for team projects as a component of software engineering courses and the need to encourage students to concentrate on the early stages of the lifecycle before producing the code. Their conclusion is that there is a need to include team projects, which will endorse the necessity to concentrate on the early process of the life cycle.

Issues in Systems Development Projects

Based on official curricula recommendations and the author's personal experience, the important issues to be considered in systems development projects, aside from the phases of the Systems Development Life Cycle include:

- Resources
- Teamwork
- Systems Integration
- Testing

- Maintenance (part of the life cycle, but often neglected)
- Meeting user requirements and their expectations
- Requirements Management
- Technical Feasibility
- Operational Feasibility
- Financial Feasibility
- Project Planning (Schedule Feasibility)

Although most of these issues are included in the IS curriculum, it is easier for students to grasp the importance of these issues by practicing them within the confines of a project in which they are specifically incorporated.

The key objective of systems development in the workplace is to achieve a working system which meets the user requirements. The key objective of a systems development project in the IS curriculum is to put into place and practice the important issues explained above.

Generation 1

The first generation systems development projects (circa 1991) were undertaken in a class of five students.

Team structure

Development teams of two or three students each were selected by students themselves.

Project source

Systems development projects were sourced from the community by the students themselves, and staff did not get involved until the project proposals were submitted. Their project proposals were verified and scaled to a manageable size by academic staff of the Department before the students began the development. The projects were small with little scope for any further development and no scope for integration with existing projects.

Systems analysis and design

User requirements were gathered by the students by way of interviews, observation and the study of existing (manual) systems and their documents. Systems Analysis was carried out using a Structured Analysis Methodology which used Data Flow diagrams and Structured Charts. Students held infrequent meetings with their clients, usually on the premises of the client. Feedback to the clients was mostly verbal. Design methodologies were still in their infancy, and the students essentially progressed from analysis through to implementation with little or no design.

Development and implementation

The projects were developed in SB and PIC Basic, which was a very successful teaching language at that time. Systems software operating systems were simple with no Graphical User Interfaces (GUI) and little incorporation of Human Computer Interaction (HCI) principles.

Deliverables

Deliverables included an Analysis Document and a User Manual as well as the working system. Assessment was carried out at the completion of the project where the final product was marked by staff members. Client input was considered when the final mark was being awarded.

Assessment

The main emphasis and therefore assessment of the systems development project was on the final working system. The documents were assessed, and the marks assigned to these counted a small percentage towards the final mark awarded for the entire project. The overall mark for the project was awarded to each team member.

Reflections:

- The students were exposed to real world projects and clients.
- The project was run over the entire year – which gave the students the opportunity to practice new skills as they were learned.
- Client expectations were not always met as they were expecting a working project that required no maintenance.
- Too much emphasis was placed on the final working product, with little emphasis on the process.
- Limited, if any Project Management was carried out.
- The requirements elicitation method was not sufficiently formalised. Meeting at the client site was not ideal, as the clients were often disturbed or called away during the scheduled meeting times.
- Teams and projects were small, which offered little challenge to the students.
- Students found it difficult to source their own projects in the community because the students are mostly not from within the local community.
- The balance of skills within teams was not always ideal, as some teams were very strong in programming while others had more team members with softer skills.

Generation 2

Class sizes had increased and the second generation of systems development projects circa (1995) were undertaken in classes of between 12 and 16 students. The curriculum had slowly evolved as well, to include more of the softer skills, which were recommended by industry. These included areas such as corporate communications, team development skills, requirements elicitation and HCI.

Team structure

Development teams of five or six students each were selected by the members of academic staff. This was done in order to spread the skills of the students more equitably. Factors that were considered while selecting team members for a team included the student's skills and ability. There were usually more "soft skilled" than "hard skilled" students. The teams were selected to include students with both sets of skills which would enable them to develop more successfully as a team

Project source

Development projects were sourced from within the community by the Department academic staff, by advertising in the local press, inviting community involvement. Students had found it increasingly difficult to source their own projects as Grahamstown is relatively small. The staff members still scoped the project sizes, which were still relatively small, stand alone projects with no integration with any existing systems.

Development approach

The Information Engineering (IE) methodology (Satzinger, Jackson, & Burd, 2007, p. 226-230) was adopted which uses Entity Relationship Diagrams, Function Structure Analysis and Interaction Analysis using Process Logic, Entity Life Cycle Analysis and CRUD matrices techniques to gather and analyse the requirements. Human Computer Interaction became an important factor as GUI interfaces replaced the old green screen systems over time. Students were expected to spend more time designing their GUI interfaces to meet user requirements. Although the analysis was carried out using structured analysis techniques, implementation was carried out using Object Oriented programming languages, such as Visual Basic and Delphi. (Satzinger, J., Jackson, R. & Burd, S. 2007)

Analysis and design

Joint Application Development (JAD) was used to gather user requirements. This workshop method required a specially designed JAD room which was a new resource acquired by the Department. JAD facilitates group participation and brainstorming which decreases the time that it takes to gather requirements, and more ideas are generated. The client is also removed from any distractions in the workplace. Students held regular JAD meetings with their clients and feedback was produced in the form of minutes of their meetings which included models of their systems so that the users could give meaningful feedback at the next meeting. (Satzinger et al., 2007, p. 138-141).

Implementation

Each team member was expected to participate in every stage if the life cycle, but in reality the programming or implementation stages of the system were normally carried out by the students with stronger technical skills. The team sizes were larger and lack of equitable member participation was a major concern. It was perceived that there were always two or three of the team members who carried out the bulk of the work, while the others appeared to play a less significant role. This observation was not always correct, as the programmers often believed that the Analysis and Design aspects of the system to be of little importance. Buddy ratings were implemented to award more favourable marks to team members. This method enables each team member to rate their own and their team members' participation in the project. Academic staff members facilitated the buddy rating and the final project mark was distributed amongst the team members based on the outcomes of the Buddy Ratings.

Deliverables

Deliverables included a Systems Requirements Specification, Systems Design Document, Implementation Document and a User Manual. Although project management techniques were encouraged and the Gantt charts included into the documentation, little attention was paid to the accuracy of the project management aspects of the development project. The focus of the project was still very much on the end product, although much more attention was paid to the Analysis and Design aspects of the project.

Assessment

For the assessment of the project, more weighting was allocated to the Analysis and Design aspects of the project, which meant that there was a greater emphasis placed on the process, and the mark weighting for the documentation was increased.

Reflections:

- The new emphasis on the process rather than the end-product ensured that the students worked more consistently on their projects.
- Client expectations were still not always met, as maintenance of completed projects was not addressed.
- Student participation not always perceived to be equitable by all the team members, and the Buddy Rating system was implemented.
- Team members did not all have the same interests and skill sets, but were expected to participate equally at every stage of the development life cycle.
- Although Project Management was implemented, not enough attention was paid to this aspect of the process.
- Industry partners were suggesting that there should be a greater focus on integration and they also showed concern regarding the lack of software testing during the development process.

Generation 3

The latest generation systems development projects (circa 2007), is being undertaken in a class of 23 students.

Features of the generation 3 systems development projects include a newly devised specification document which includes Critical and Specific outcomes. (Table 1).

Team structure

- The team structure has been modified to include five distinct development roles, which are:
- Project Manager
- Analyst (Requirements Elicitation and Analysis)
- Designer (Logical and Physical Design)
- Implementer (Programming)
- Software Tester (testing at all levels of the systems development life cycle)

The projects teams are selected by members of the academic staff. Each project team consists of one Project Manager, Analysts, Designers, Implementers and Testers. Each team member is assigned to two of these development roles, according to their individual strengths and/or preferences. This means that there could be between 10 or 12 allocated roles in each team of 5 or 6 team members respectively. This is in line with the “Specific Outcomes a, b and h”. (Table 1).

An example of the roles of a team of six members is the following:

- Team member 1: Project Manager and Analyst
- Team member 2: Analyst and Tester
- Team member 3: Designer and Tester
- Team member 4: Designer and Implementer
- Team member 5: Analyst and Implementer
- Team member 6: Designer and Implementer

Table 1

"OBE Specification Document for Honours System Development Project : 3-4" (McNeill 2007)

Outcomes

2.1. Critical Outcomes

Students will be expected to:

a. identify and solve problems
b. work in a team
c. organise and manage him/herself
d. collect, analyse and evaluate information
e. communicate effectively
f. use science and technology
g. recognise problem solving contexts
h. reflect on and explore effective learning strategies
i. participate as a responsible citizen
j. be culturally and aesthetically sensitive
k. explore education and career opportunities
l. develop entrepreneurial opportunities

2.2. Specific Outcomes

At the conclusion of the module, students should be able to:

a. Carry out the responsibilities and perform the duties and tasks/functions of at least two (2) specialisation roles.
b. Use appropriate tools and techniques within their specialisation roles.
c. Investigate, explore and use new technologies and tools with support from staff/specialists.
d. Understand and integrate the various life-cycle stages.
e. Interact with the other specialisation roles.
f. Interact productively and creatively with the clients and users of the system being developed.
g. Develop an information system incrementally and iteratively.
h. Work in a team of not more than six (6) students.
i. Contribute to a working system that meets user requirements.
j. Present and promote their system to users.
k. Develop and present appropriately tailored documentation.

Project source

A suitable systems development project is to be sourced by the staff from the Department. The project should be large enough to be divided appropriately into three or four sub-projects. Each development team is responsible for developing one of the sub-projects. This approach forces the students to consider integration and interface consistency issues during the development process. The role allocation within the teams also encourages students to participate and to be accountable for their specific aspects of the system. Although it is believed that buddy ratings will not be necessary, due to the nature of the new team structure, this will be monitored carefully and could be implemented if necessary, on completion of the project.

At the start of the project all the teams will work together to gather the high level requirements for the entire system. These requirements will be analysed and the team members, will divide the system into three or four logical sub-systems. At this stage, each team will adopt their allocated sub-system to continue the development in an iterative and incremental manner, discussed below.

Development approach

The Rational Unified Approach (RUP), using the Unified Modelling Language (UML) has been adopted as the development approach. The RUP approach is iterative and incremental in nature, and teams are expected to develop their systems over several iterations. Each iteration covers the full systems development life cycle. (Stumpf & Teague, 2005, p. 31-65).

Microsoft's Visual Studio Team Suite (VSTS) has been identified as the working environment for the project, and students are expected to adopt this environment which encourages team work and integration. Other software tools that are being used are Rational Requisite Pro and Rational Rose.

Analysis and design

UML Analysis and Design modelling techniques are used for the Analysis and Design stages of the Systems Development Life cycle (Stumpf & Teague, 2005).

During the initial phase, Iteration 0, each team will elicit the high level requirements for their system and produce a specification document which includes a high level Use Case model revealing the use cases for their systems. This high level specification model is used to divide each system into three or four incremental iterations. Once the incremental stages for each iteration has been identified then the analysis, design, implementation and testing for each iteration will be carried out.

JAD sessions, using techniques which include brainstorming, "Six Thinking Hats" (De Bono, 1999), and other elicitation methods are to be used to gather requirements. Requirements Elicitation plays an important role in the process and a great deal of emphasis is placed in this stage of the life cycle.

The Analysis activity is performed for each iteration of the system's development project. Analysis activities include:

- Requirements Elicitation: Requirements Specifications, a Context Diagram, a high level Use Case Diagram and Class Diagrams are produced.
- Systems Analysis: Use Case Narratives, Class Diagrams, Activity Diagrams and Sequence Diagrams.

The Design activity is performed for each iteration of the system's development project. Design activities include:

- Systems Architecture Design: Architectural Layers are developed including network, security and web specifications.
- Program Design: Analysis use cases and narratives are elaborated and extended. Collaboration diagrams are produced for each system operation identified in the uses cases and a Design Class Diagram is fully developed. Full narrative specifications are defined for each of the class operations.
- User Interface Design: Prototype screens and reports are developed around a generic specification and Window Navigation diagrams are created.
- Database Design: Relation database models are developed. Attributes are framed (specified) and backup regimes are specified.

Project management

Given that team members each have two different roles in the lifecycle, project management plays an important part in allocating resources. The project manager for each team is responsible for allocating resources throughout the entire process. Some of the team members are have the responsibility to ensure that proper integration takes place. The deliverables of Project management include a Work Breakdown Structure to define the scope of the project, and a Project Network Diagram, to indicate the critical path, produced by MS Project (2003). Project managers are required to produce a Gantt-Chart, to represent associated project resources to identify resource problems and suggest appropriate solutions.

Development & implementation

VSTS is the chosen development and implementation environment. VSTS is an enhancement to the Microsoft .NET platform that facilitates team development. The department utilises mainly the source control and co-ordination facilities provided by the system. A team project is created that provides source control which allows for code to be managed via check-in policies, and provides the ability to roll back to previous versions of code. VSTS provides development facilities for designers, coders and testers, with automation of associated processes e.g. test stub and harness generation, daily builds, bug tracking and commenting, website publishing, logical and physical architecture diagrams, deployment diagrams and packages. All the coding is currently done in C#.

VSTS provides a project portal which enables students to publish documents associated with the project on-line. It also allows for basic project management e.g. the allocation of work-items (or tasks) to individuals in the project. All users (including staff) have controlled access (based on rights) to the portal – as designated by the administrator.

Visual Studio and VSTS leverages industry best practices in a controlled manner to expose students to the process of software development as an iterative and value added process with designated deliverables in a robust flexible environment. The adherence to standards and practices opens up the opportunity to roll projects over from one year to the next, leading eventually to a deployable, maintainable and usable system.

Deliverables

The following deliverables are produced by the students for assessment and feedback during various stages:

- Requirements Specification Document is produced after the completion of Iteration 0. This document includes the users' business requirements for the system and provides a breakdown of what will be included in each of the iterations to follow.

On the completion of each of the subsequent iterations and completion of the entire project, the following documents are provided as deliverables:

- Project Management documentation
- Systems Requirements Specification documentation
- Design Documentation
- Implementation documentation
- Testing documentation
- Demonstration of the system for assessment

Assessment

Formative assessment is conducted throughout the entire process. The teams are required to present their systems, on completion of their iterations, to the users and staff. The implementation of the system is assessed during their presentations and feedback provided to the students. Team members from all the teams attend the presentations, which assists in integration and consistency.

Marks are assigned to all documentation and presentations at all the stages of the process throughout the entire project, which ensures the balance of focus between the process and the completed project.

Conclusion

This paper discussed the issues experienced and addressed while integrating a systems development project into IS Honours curriculum at Rhodes University. The project structure, team dynamics, and assessment, have been continually revised to improve on the process. The current generation has introduced several new aspects to the running of the projects, which include a larger system being broken down into subsystems, the team members being allocated roles within the development team, and the use of an iterative approach to development which facilitates ongoing assessment.

References

Avison, D. & Fitzgerald, G. (2006). *Information systems development methodologies, techniques & tools.* McGraw-Hill.

Anewalt, K., Polack-Wahl, J.A., Beidler, J., & Smarkususky, D.L. (2003). Group projects across the curriculum. *Journal of Computing Sciences in Colleges, 19*(2); 232 – 237. Publisher: Consortium for Computing Sciences in Colleges.

De Bono, E. (1999). *Six thinking hats*. Penguin - Psychology/Business.

Gorgone, J.T., Davis, G.B., Valacich, J.S., Topi, H., Feinstein, D.L., & Longenecker, Jr, H.E. (2002). IS2002 model curriculum and guidelines for undergraduate degree programs in information systems. Retrieved 25th April, 2007 from http://www.acm.org/education/IS2002.pdf

Gorgone, J.T., Gray, P., Stohr, E.A., Valacich, J.S., Wigand, R.T. (2006). MSIS 2006 model curriculum and guidelines for graduate degree programs in information systems. *Communications of the Association for Information Systems, 17*(1).

McNeill, J.B. (2007). OBE specification document for honours system development project. Unpublished manuscript, Department of Information Systems, Rhodes University, Grahamstown, South Africa.

Satzinger, J., Jackson, R., & Burd, S. (2007). *Systems analysis & design in a changing world* (4th ed.). Thomson Course Technology.

Sherrell, L.B. & Shiva, S.G. (2005). Will earlier projects plus a disciplined process enforce SE principles throughout the CS curriculum? *ICSE '05,* May 15-21, 2005, St Louis, Missouri, USA. ACM.

Smarkusky, D.L. & Smith, H.H. (2004). Team projects throughout the curriculum: Course management, teaching initiatives and outreach. *Journal of Computing Sciences in Colleges, 19*(5); 119 – 129. Publisher: Consortium for Computing Sciences in Colleges.

Stumpf, R.V. & Teague, C.T. (2005). *Object-oriented systems analysis and design with UML*. Pearson Prentice Hall.

Torngren, M., Grimheded, M., & Adamsson, N. (2007). Experiences from large embedded systems development projects in education, involving industry and research. *ACM SIGBED Review, 4*(1), 55 – 63. Publisher: ACM Press.

Biography

Mrs Lydia Palmer, BComm, MBA (Rhodes) is a lecturer in Information Systems at Rhodes University in Grahamstown, South Africa. She is involved in teaching Information Systems at all levels. After studying for her BComm at Rhodes in Computer Science, she worked as an Analyst in Industry for 6 years. She started lecturing in 1984 and obtained her MBA in 2003. Her research interests include: Internships in Education; Requirements Elicitation; Systems Analysis, Software Testing and HCI. You can email Lydia at l.palmer@ru.ac.za.