9. TOWARDS A STANDARD MONITORING SYSTEM FOR COMPUTER SCIENCE STUDENT PROJECTS

R. K. Mazhindu-Shumba
G. T. Hapanyengwi
Computer Science Department, University of Zimbabwe, MP 167, Harare, Zimbabwe

ABSTRACT

This paper describes two types of Computer Science projects that students get involved with. It then proposes a monitoring system for projects. The proposed system has been developed and used in the Department of Computer Science at the University of Zimbabwe. The monitoring system involves (a) an effective reporting structure, namely end of task reports and fortnightly reports, (b) three oral presentations, (c) a final system demonstration. The monitoring system has been used for the past three years and has been found to be very effective.

Keywords
Computing, Assessment, Information Technology, Software Engineering

---

The original version of this chapter was revised: The copyright line was incorrect. This has been corrected. The Erratum to this chapter is available at DOI: 10.1007/978-0-387-35393-7_22
P. Juliff et al. (eds.), *Educating Professionals for Network-Centric Organisations*

1. Introduction

The Department of Computer Science at the University of Zimbabwe offers structured Computer Science undergraduate courses to students in the Faculties of Commerce, Engineering, and Science. Among the courses offered from second year level to fourth year level is a Project. A Project is a major Computer Science course in which students are expected to develop and demonstrate both powers of initiative independent thought and a professional standard in report submission. A Project addresses real world problems using Computer Science skills at the student's disposal. Projects are either hardware projects or software projects.

In this paper we describe the two major types of projects undertaken by students, the developer-solicited project and supervisor-assisted project, and then propose a monitoring system[1] for the projects. The proposed monitoring system has been developed and implemented/used in the Computer Science Department of the University of Zimbabwe by computer science project supervisors. A supervisor is an academic member of staff to whom the developer is answerable. Our experience of using the system has shown that it is a very effective monitoring system which when properly implemented benefits both the student and the supervisor.

2. Background To Projects

Student projects are of two types: student-solicited and supervisor-assigned. Student-solicited projects result from either a vacation-attachment or project initiated by a client. We shall refer to these as vacation-oriented and client-oriented projects respectively.

A vacation-oriented project might be part of a more comprehensive project. In such a case a student has to understand the more comprehensive project. The comprehensive project becomes part of the environment of the project. The problems related to the vacation-oriented projects are that either the student is tempted to implement what has already been implemented or the student may feel that scaling down of an already developed system might constitute a project. In such a case, students are advised to seek clarification from a more experienced fellow-student or a prospective supervisor. These two should help in ascertaining what should pass as a sufficient project.

Client-oriented projects solve specific problems, which are not necessarily related to other aspects of the day to day function of the client's business. Depending on the calibre of the client, it might be up to the student to extract a manageable subset of the problem at hand.

Student-solicited projects usually end up with two supervisors; the internal supervisor who is an academic member of staff and the co-supervisor from the client organisation who is responsible for the project. It is the student's responsibility to work hand in hand with both supervisors.

---

[1] This monitoring system is specific to software projects.

Supervisor-solicited projects are based on problem areas, which are workable within the given time and resource limits. The student normally relies on the supervisor's experiences in determining the resources. However, the student still needs to determine appropriateness of the project with respect to her abilities and her interests.

3. Team Projects

Projects may be done individually or in teams. Teams are usually made up of students enrolled in the same programme. The need for teams could be due to the size of the project or to the diverse skills required for a given project. Our experiences of supervising projects show that properly coordinated teams tend to produce better quality projects. This, we feel, is attributed to the sharing of ideas among the team members. In fact team projects allay student fears. Students doing a project for the first time (second year level students) are not sure of what is expected of them. The fact that the students are expected to produce a software system is, on its own, scary to most second year students. From experience, students are more comfortable the time they learn that a project is a team project. Although teamwork helps make students more comfortable, there are a number of issues to be considered by students in choosing a team project:

- Choice of the team members
- rapport within the team
- accountability of the team members
- Reporting structure and authority within the team.

4. The 1997 Practical Projects In Computer Science Department

1997 saw a proliferation of projects of which about 60% were supervisor-assigned and about 40% were student-solicited. Of the 60% supervisor-assigned projects, 20% were hardware projects and the rest were software projects. Software projects require that students go through the whole software development process.

A software development process may be viewed as comprising requirements analysis and definition, system design, program design, program implementation, unit testing, integration testing, system testing, system delivery and maintenance. The requirements describe what is required of the system. Once the requirements are defined, a system design is created to meet the specified requirements. The system design shows the user what the system will look like from the user's perspective. The user then reviews the design. When approved, the overall system design is used to generate the designs of the individual programs involved. When programs have been written, they are tested as individual pieces of code (unit testing) before they can be linked together. An integration test is then done. The final testing phase, called system testing, involves a test of the whole system to

make sure that the functions and interactions specified initially have been implemented properly. At last, the final product is delivered. Software projects require students to design and implement software systems.

Some of the projects done during the 1997 academic year include; A Pharmaceutical Monitoring System, A Tourist Market Research Tool, Computer-Aided Educational Tools for Primary and Secondary Schools, A Car Hire System, Help Desk Automation System for the Computer Centre, A Till Reconciliation System, and A Project Recording and Consultancy Tracking System. We shall briefly describe two of the projects.

One of the Computer-Aided Education tools developed is the "Indigenous People and their Life Style System". Besides producing a map of Zimbabwe showing the geographical distribution of tribes, the system allows for the viewing of the following: the languages, the crafts, sacred places and taboos, beliefs associated with the tribes, the staple foods and the agricultural activities for the respective tribes. The main benefit of such a system is that it allows for quick and easy access to information about lifestyles and the indigenous people. Such information is invaluable in teaching and preservation of History. Since the system gives the proportion of each tribe relative to the whole Zimbabwean population, this gives policy makers a good planning platform. For example, the proportional distribution of languages gives an insight to the amount of literature to be produced.

A Business Studies Department lecturer solicited the development of a Tourist Market Research Analysis tool. The developed system addresses one problem facing many disciplines today, that is the fragmentation of information. Tourist information is so fragmented that it cannot be used for any specific purposes. The Tourist Market Research Analysis tool captures and stores data pertaining to tourists visiting Zimbabwe. The data stored include; the profile of the tourist, tourist products (places of interest), gaps with respect to products being offered (improvements to present tourist offers by the Zimbabwe Market), food and beverages, accommodation, transport used, entertainment, and curio (types of curio bought and the price). The data is then analysed for example by country to show the number of tourist that visit Zimbabwe from a given country, their sex, expenditure pattern and average income. Narrative reports graphs and tables are then given to present the results.

5. A Proposed Monitoring System

As mentioned in Section 3 of this paper, the main tasks of a software Practical project include problem definition, feasibility study, project planning, requirements analysis, design, implementation, and then finally the testing. These tasks are performed over a period of about seven months. Our experience of supervising projects has shown that students, who work steadily following the proposed monitoring system, tend to produce better quality projects. However, for the other students who strive to do the whole project within a "month" before the submission date, usually stress themselves up and in the end do not perform as well.

We therefore are proposing a number of measures to monitor the progress of the students' projects. The proposed measures have been implemented in the

Department of Computer Science over the past three years and have been found to be very effective. The measures include:

- An effective reporting structure: end of task & fortnightly reports.
- Three oral presentations
- Final system demonstration.

6. An Effective Reporting System

6.1 Fortnightly Reports

The student should maintain fortnightly written reports, in a folder, from the day of commencement of the project. Such a reporting structure assists the student with a systematic and structured record-keeping format. Figure 1 gives the layout of a typical fortnightly report.

Project Title:_____
*Developing Team:*_____

1. The Plan for the coming two weeks.
2. Work done in the previous two weeks.
3. Problems faced during the previous two weeks and their solutions.
4. The current status of the project.
5. Any other relevant information.

Figure 1: Fortnightly report for a project

We encourage students to submit both previously corrected documents and current documents for discussion. This assists with keeping track of the progress of the project. It also makes discussions non-repetitive. The reports should also constitute meetings between the supervisor and the student. Reports for discussion should be handed to the supervisor at least forty-eight hours before the scheduled meeting.

6.2 End of Task Reports

As we noted above, students are expected to perform certain tasks. We expect that at the end of each task, a progress report be produced. In all, each student should produce the following reports:

- An initial project document.
- A specification document.
- Feasibility/scope and objective report.
- IEEE SPMP report
- Analysis report.

- Design report.
- System assessment report.

Within two weeks of the start of the project, we expect an initial project report. At this juncture the student ascertains a few points with regards to the choice of the project and the development team. Figure 2 is a sample of the report structure for the initial project document.

*Project Title:*_____
Developing Team: _____

1. Brief Summary of the problem
2. Non functional requirements
   Any materials and resources needed to understand the fuzzy aspects of the problem
3. Summary of adequacy of the project
   3.1. Challenging aspects of the project
   3.2. Progress control mechanisms

Figure 2: Initial project document

The second report that the students have to produce is the requirements specification report. This report specifies the system requirements in a more formal structure. A typical structure is given in Figure 3. On average this report is produced a month after the start of the project.

After performing the requirements and specifications the student has a clearer picture of what the project entails. The student should then perform a critical analysis of herself with respect to the task at hand. This analysis is to determine whether:

- There are aspects for which the student is not adequately prepared. In this case the student has to work on acquiring the relevant skills.
- The whole project is too much to handle in which case whether there is need for expanding the developing team or whether it is prudent to abandon the project all together.
- The whole project is not as comprehensive in which case there is a need to expand the requirements or trim down the size of the team.

*Project Title:*_____
*Developing Team:*_____

1. Introduction
   1.1. Need for the new system
   1.2. Brief description of how the system fits into the business
2. Functional Requirements
   2.1. Description of services provided to the user
3. Boundary of the System
   3.1. Description of the environment
4. Non-functional Requirements
   4.1. Relevant Costing
   4.2. Hardware and software requirements
   4.3. Manpower requirements
5. System Evolution
   5.1. Assumptions on which the system is based
   5.2. Anticipated changes due to hardware evolution
   5.3. Anticipated changes due to software evolution
   5.4. Anticipated change in user needs
6. Prototyping report
   6.1. Goals for prototyping
   6.2. Functions prototyped
   6.3. Prototyping results
7. Glossary

Figure 3: Specification Document

Upon completion of the Specification Report, we expect the student to demonstrate the developed prototype system. This helps the supervisor to confirm the student's progress.

At this stage of project development the student should have a firm direction as to what needs to be implemented. To reaffirm their understanding the first oral presentation should be given. This presentation should give a clear and concise summary of the project, highlight the cost and the benefits, and present the results and recommendations of the feasibility study.

At the end of the feasibility study, a report has to be produced. A typical structure of the report is given in Figure 4.

Though planning is an ongoing process, it is at this stage that a project planning report is compiled. Figure 5 is an example of a typical project planning report. The report is adopted from the IEEE-SPMP.

*Project Title:*_____
*Developing Team:*_____

1. Executive summary: A clear and brief summary of the feasibility study:
   1.1. key sources of information
   1.2. alternative solutions considered
   1.3. highlights of the costs and the benefits
   1.4. results and recommendations from the study
2. Appendix:
   2.1. Charts
   2.2. Graphs
   2.3. Interview lists
   2.4. Selected interview summaries
   2.5. Memos and notes
   2.6. References

Figure 4: Feasibility/Scope and Objective Report

*Project Title:*_____
*Developing Team:*_____

1. Introduction
   1.1. Project overview
   1.2. Project deliverables
   1.3. Reference materials
   1.4. Definitions and acronyms
2. Project Organisation
   2.1. Project responsibilities.
3. Managerial process
   3.1. Management objectives and priorities
   3.2. Risk management
   3.3. Monitoring and Controlling mechanisms
   3.4. Staffing plan
4. Technical process
   4.1. Methods, Tools and Techniques
   4.2. Software documentation
   4.3. Project support functions
5. Work Packages, Schedules and Budget
   5.1. Work packages
   5.2. Dependencies
   5.3. Resource requirements
   5.4. Budget and resource allocation
   5.5. Schedule

Figure 5: IEEE SPMP Report

At the end of the analysis stage, an analysis report is produced. The contents of the analysis report are as illustrated in Figure 6.

*Project Title:*_____
*Developing Team:*_____

1. Presentation of a model[2] for the whole system.
2. Relevant data dictionary
3. Presentation of process logic using one of the tools

Figure 6: Analysis report

The system's analysis and design at this stage needs external assessment. Such assessment can be achieved through an oral presentation. The presentation should be targeted at a diverse audience to include the developer's peers, the user (if the user is competent to evaluate a project at this level), the supervisors, and other informed outsiders. This oral presentation should include the following aspects:

- A brief description of the requirements definition and specification of the project
- The analysis results: justification for choice of analysis tools
- Summary of the project plan
- The design: System Relationship Diagram (SRD), and structure chart
- Problems encountered

*Project Title:*_____
*Developing Team:*_____

1. Design of sub-units
   - 1.1. Algorithm for each sub-unit
   - 1.2. An SRD for the sub-units
   - 1.3. Structure chart for the sub-units
   - 1.4. Flowcharts for any complex logic
   - 1.5. An assessment of quality of the algorithm
2. Design of user interface
   - 2.1. Forms
   - 2.2. Menus
   - 2.3. Reports
   - 2.4. An assessment of quality of user interface

Figure 7: Design report

[2] A **model** can be presented using either a **DFD** or an **ERD** or a combination.

At the end of the systems design stage a design report is produced. The contents of the design report are as illustrated in Figure 7. This report can also be used in the oral presentation.

At the end of the implementation stage a system assessment report has to be produced. In this report an outline of the various parameters of assessment is made. Figure 8 outlines the structure of the system assessment report. What have to be brought out by this report are the strengths and the weaknesses of the system. Thus, in assessing any one of the aspects it should be clear as to how well the system handles that specific aspect.

*Project Title:*_____
*Developing Team:*_____

1. Adequacy
2. Portability
3. Reusability
4. Adaptability
5. Environment chosen.

Figure 8: System assessment Report

The final demonstration is a one-time affair. Interim demonstrations can be arranged any time. The developer should not only use the supervisor as the tester of the system. In fact, as part of the developer test plan, the developer should have a panel of people whose job is to test the system. It is always easier to rectify an error, which is discovered earlier in the development process.

Unfortunately it is not always possible or advisable to get the user involved in testing an unfinished product. The developer, however, has to simulate the needs of the user as closely as possible. A very productive thing to do is to have one of the developer's classmates play the devil's advocate every so often. The developer might be amazed as to how much help a fellow student might give.

6.3 The electronic report folder

In an institution with an Intranet and a computer network, communication between the supervisor and the project team can be significantly enhanced. A project folder is created for each of the projects. Each folder is accessible through a password known to the supervisor and the development team. The folder is located in the workspace of the developing team.

The structure and organization of the folder is akin that of the paper folder. Correction of a report is maintained by keeping a version of the given report. The supervisor and the project team are each allocated a different color or type font to use. In this way, distinguishing the various contributions is straightforward. Drawbacks with such a system include documents needing intensive correction and limited typing capabilities on part of the supervisor.

7. Conclusion

This paper has explained two types of projects undertaken by Computer Science students: supervisor-solicited and student-solicited projects. Student-solicited projects can either be vacation-oriented or client-oriented. Regardless of the type of project, a student has to fully understand the subject area for any chosen project. In some cases, it might be necessary that the student carry out relevant research in order to acquire the pertinent knowledge.

The tasks for a given software project include problem definition, requirements analysis, feasibility study, project planning, requirements analysis, design, implementation, and testing. Our experiences of supervising projects has shown that there is need for a monitoring system. We have therefore proposed a number of measures that may be put in place to monitor the performance of students. The measures include an effective reporting structure: fortnightly reports and end of task reports, three oral presentations and a final system demonstration.

The standardisation of the monitoring system is an on-going exercise. We feel that this can be extended to the grading of the project. There is actually some work in the Department of Computer Science on the standardisation of the grading system for Computer Science projects.

8. References

Schach Stephen, R. (1993) *Software Engineering*, Second Edition. Irwin.

Britton, C. and Doake, J. (1993) *Software System Development.* The McGraw-Hill International Series.

Somerville I. (1995) *Software Engineering*, Fifth Edition. Addison Wesley.

9. Biography

Gilford Tapera Hapanyengwi is the Director of Computer Center at the University of Zimbabwe. He holds a PhD and MS degree in Computer Science from Vanderbilt University in the United States of America and a BSc degree in Computer Science and Mathematics from the University of Zimbabwe. Gilford's research interests are in Knowledge Discovery in Database Systems, Geographical Information Systems, Software Engineering, and Computer Education.

Rose Kudzai Mazhindu-Shumba is the Chairperson of the Computer Science Department at the University of Zimbabwe. She holds a PhD in Computer Science from Birmingham University in England, a M.Sc. in Computer Science from University of Manchester in England, and a Bachelor's degree in Computer Science and Business Studies from the University of Zimbabwe. Rose's research interests are in Software Engineering, Object Oriented Systems, Operating Systems, Programming Languages, and Computer Education.