Information and Guidelines for Computer Science Projects (COMP SCI 4ZP6)

Course Coordinator: I. Bruha

Department of Computing and Software McMaster University

Revised on Sep.2009

(The format of this report follows the requirements for Computer Science projects.)

Table of Contents

1 GETTING STARTED 1

2 HOW A PROJECT IS DONE 2
- 1.1 Analysis 2
- 1.2 Extended Project Proposal 2
- 1.3 Implementation 3
- 1.4 Validation/Testing 3
- 1.5 Analysis of Results 4
- 1.6 Writing the Report 4
- 1.7 The Supervisor's Role 4
- 1.8 Failure to Complete the Project 5

3 THE PROJECT WRITE-UP 6

4 TYPING AND REPRODUCTION OF THE REPORT 8
- 4.1 Typing 8
- 4.2 Number of copies 8
- 4.3 Deadlines 8

5 EVALUATION 10
- 5.1 Marking Scheme 10
- 5.2 Evaluation of Project Work 10
- 5.3 Evaluation of the Project Report 10
- 5.4 Oral Examination 11

BIBLIOGRAPHY 12

APPENDIX I: Sample Title Page 13

APPENDIX II: License to McMaster University 13

1 GETTING STARTED

In the final-year project course, CS4ZP6, a group of two to four students works under supervision on a chosen topic for two terms. The main purpose of the project is to expose the student to a large problem whose solution involves both teamwork and organized thought. The project course gives students a chance to become familiar with a part of Computer Science that they find interesting and the opportunity to gain employment-related experience.

There are three steps involved in getting started on a project:

1. The first step is to form a group of two to four students with broadly similar interests. As you will be working with your partners for two terms, select people you can get along with.

2. Each group then looks for a topic in its area of interest. You may consult the list of project proposals, available on our web site, contact any professor of our department, contact any external company to find an external supervisor, or come with your own ideas and look for a supervisor.

   Copies of the top three projects from previous years are shelved with theses in the basement of the Thode Library. A new project may continue an older topic, or repeat it again with significant improvements. A supervisor should help you with the final choice of a project topic. If a suitable project proposal cannot be agreed upon, you may have to change to another supervisor.

   Many projects arise in an application area and these usually come from discussions between a group and a faculty or a staff member of another department. It is also possible for projects to be carried out and supervised off-campus; for example, successful projects have been completed at Dofasco.

3. Now that you have a topic and a supervisor you must complete a *Record of Proposed Project Form* available on our web site, and contact the course coordinator.

   If your supervisor is not a faculty member of our department, a co-supervisor (liaison) from our department will be required (s/he will also serve as an examiner). Contact the course coordinator.

2 HOW A PROJECT IS DONE

There are six main phases in doing a project and these are described below. Brooks (1995) is an interesting book about projects.

2.1 Analysis

Most project proposals are very imprecise and open-ended. In the analysis phase, the problem is thoroughly examined and a precise detailed specification drawn up. This phase is vital and must not be skimped -nothing can rescue a poorly analyzed project. It is normally the case that a literature search is a vital complement to your group discussion and thinking. A preliminary schedule of activities should be drawn up at an early stage so that a target date for the completion of the analysis is known. The schedule will be finalized once the analysis is complete.

Frequent meetings with your supervisor or liaison (if you have an external supervisor) are necessary and it is recommended that such meeting be held on a regular basis, usually every two weeks.

2.2 Extended Project Proposal

In this phase algorithms required for the solution of the problem are devised and described in detail, possibly using a pseudo-code. In particular, examination of previous project reports in the same or related areas is important.

The results of Phases 1 and 2 are written up and submitted in the form of an *Extended Project Proposal (EPP).* The proposal should be between six and ten pages in length and should explain why the work was needed and who are the intended users *(motivation);* what has to be done to achieve the end results and what tools are required to do the work *(objectives);* how the above objectives will be reached and how the tools will be used to do this *(methodology);* lastly, when the separate stages of the work will be completed and when the constructed system will be tested *(timetable).* While no specific form is required for this course, selecting an appropriate design methodology and formal specification will be viewed positively.

The EPP should have the following sections:

- **Title Page**

- **Table of Contents**

- **Introduction**
  The Introduction should tell the reader
  a. What the report is about: Define or describe the subject area; briefly state what is available
  b. Why the report was written: Set the scope of the work; describe who is going to use the product of the project; state how the users will benefit by your work

- **Body of the Report**
  The Body of the Report should provide the reader with
  a. A statement of the problem: This is the key section; should provide full background
  b. Proposed system or methodology
  c. Facilities and equipment required: List hardware and software you intend to use; describe how the work is going to be divided among group members
  d. Duration or schedule: List milestones so that progress can be checked

- **Conclusion**
  The Conclusion should be short and should provide the reader with general comments on the proposal.

- **Bibliography**
  List all items referenced, including books, journal articles, programs, manuals, lecture notes, private communications from your supervisor, etc.

The *Extended Project Proposal* will be graded by the course coordinator on appearance, content, and standard of English. Its contribution to the final mark is 10%.

2.3 Implementation

The algorithms developed in the design phase are coded in the appropriate language, and component modules tested. A well-designed user interface is of great importance to ensure that your program is well received by its intended audience. On the other hand, there may not be enough time to produce anything useful if too much effort is spent in developing an interface from scratch. It is essential that you select an appropriate interface package and obtain a legal copy of it, so that you can concentrate on the primary goal of your project.

A prototype of your program must be demonstrated to your supervisor (see deadlines). The prototype should show the user interface and the overall structure of the final program. If there is a significant database component, enough of the database should be populated to test that the structure is sufficient to support the application. If there is a complex algorithm involved, providing a proof of convergence would be useful and very impressive.

2.4 Validation/Testing

A variety of techniques can be brought to bear on the question, "Are the computed results correct?" The processing of some carefully chosen data will be necessary to detect typing errors, at least. The effort required for validation can be greatly reduced by taking pains to demonstrate the validity of the algorithms during the design phase. In your project report you should include a description of the tests performed and the test data used.

There are several testing methods; you should apply some of them:

1. *White Box Testing* tests components while isolated from the rest of the system. Because it tests components outside of the entire system, they can only supplement the Black Box tests, not replace them.
2. *Black Box Testing* is carried out on the entire system to ensure that each component performs together as expected. It is generally the most significant phase of testing and should uncover most programming errors; therefore, it is mandatory.
3. *Acceptance Testing* is the final testing conducted by the supervisor; it ends with his/her acceptance that the project is complete and met the expectations. It, in fact, simulates a commercial environment. It may be necessary if the supervisor is not familiar with the software development projects.
4. *Regression Testing* is performed when an existing system is modified or enhanced. Its focus is to ensure that the components that were not modified still work as they did before the modification. It is mandatory only if you modify your existing system.
5. *Code Review* is an optional manual review of some source code by a group member other than the author of the source code. Thus, one can identify coding errors such as loops that may fail to terminate. It may be also used to assure the clarity of the source code documentation.
6. *Stress Testing* is an optional testing that determines the ability of the system to cope with intensive use, e.g., processing extremely large databases.

7. *System Testing* is another optional testing that reveals if the new system has a negative impact on its environment; it identifies any incompatibility with existing hardware and software, problems caused by the installation, and the like.

2.5 Analysis of Results

For some projects, the final product is a reliable efficient program and analysis is then absent. For others, the project is a means of processing data and the analysis of its output is paramount. In such cases the schedule should allow time for data collection and processing and also for any subsequent analysis of the output.

2.6 Writing the Report

When writing a report it is important to bear in mind the function of the report. As far as the project course is concerned, the primary function of the report is to describe the project and its products so that any reader with some knowledge of computer science can understand it. Do not assume that the reader is familiar with the topic of your project. Therefore the project report is primarily a managerial report and secondarily a maintenance manual (particularly the Appendices) .

The report should be written in stages as work progresses throughout the year, although the month of February should be reserved for intensive word processing and formatting of the document. A final draft copy in publication-quality form must be handed to your supervisor (see deadlines). This will leave you time to make all of the changes and corrections required by your supervisor.

2.7 The Supervisor's Role

The supervisor plays four roles.
1. The first role is that of a *manager*. You should set up a schedule for your project in consultation with your supervisor and establish a number of milestones. Any problems with timing should be immediately discussed with your supervisor.
2. Your supervisor should also be viewed as the *consumer* of the system produced by the project. Any major changes or decisions in the project should be discussed with your supervisor.
3. The third role played by your supervisor is that of a *consultant*. Students should feel free to approach their supervisor with any problems they are having with their project. Your supervisor need not and, indeed, may not solve the problem, but might suggest a new approach to its solution.
4. Finally, when the project is complete, the supervisor participates in its *evaluation*.

You are responsible for completing your project, not your supervisor. Supervisors volunteer their time to help you, so treat them with respect.

In the case of a group with two supervisors, i.e. one external and one departmental co-supervisor (liaison), please be sure to provide a draft of your write-up to both supervisors in time for constructive assistance. If the external supervisor is an end-user (client) who does not understand computer science, then the departmental co-supervisor (liaison) is helpful in designing and choosing the right tools.

One important point to note is that the course coordinator has the last word with respect to the final appearance and content of the project write-up.

2.8 Failure to Complete the Project

Occasionally, a team may not complete the project as originally specified. This may arise from many causes: underestimation of time needed, hardware problems, system problems, illness of one of the team members, etc. Should this happen to your team, do not panic, but discuss the problem immediately with your supervisor and the course coordinator – the situation might be saved by making the project smaller.

3 THE PROJECT WRITE-UP

The suggested format for a project write-up is outlined below. In some cases the nature of the project may dictate another format. If this is the case, you should discuss this with your supervisor.

- ***Title-page*** is not numbered (a sample page is included at the end of this booklet).

- ***Abstract*** of one page should summarize the main points of your work. This page is to be numbered ii .

- ***Table of Contents, Diagrams, and Tables*** contains this material should be numbered starting iii .

- ***Chapter 1***. The page numbering starts at Page 1 . The chapter should give the analysis of the project, i.e. the reason *why* the problem arose in the first place (motivation), *what* are the issues involved and the setting of the problem (environment), and *how* you proceeded towards a solution (methodology). The Extended Project Proposal, which must be submitted to the course coordinator (see deadlines) can be a preliminary version of Chapter 1. It should also describe the overall program layout. Make sure that each member's contribution to the group project is carefully identified here.

- ***Chapter 2*** contains the design, i.e. an overview of the proposed solution and the work that was done. It should provide an expanded version of *how* the project was tackled and a description of *what* was done to solve the problems arising in the project.

- ***Chapter 3*** discusses the implementation of the algorithms used in the project. Only high-level descriptions are needed. If it turns out to be necessary to introduce details, then a top-down approach should be used in the exposition that gives the reader an understanding of their significance and motivates interest. More detailed specifications should be relegated to Appendix I. Make sure that there are enough pointers in this chapter and Appendix I to the code contained in Appendix II.

- ***Chapter 4***. This verification chapter analyses and summaries the results obtained. Concluding remarks should clearly state what goals were met, where compromises had to be made, and what was left out of the proposals made at the time the Extended Project Proposal was submitted. In the discussion of the solution, those methods tried but found unsuitable, and those thought of too late for testing, should be mentioned. An evaluation of the level of success attained in your work should also be made here. Finally, suggestions for future work and improvements that could be made in the project are also important.

- ***Bibliography*** contains a list of articles, reports, theses, and books consulted during the project work. The titles of books should be in italics. Titles of magazine or journal articles should not be italicized, but should be placed in double quotation marks. Journal or magazine names should be in italics. A sample bibliography is given at the end of this report.

- ***Appendix I***. Its part 1 is a users' guide; sufficient information should be provided to explain the initial installation of the program and its subsequent use. Part 2 must contain procedure names and function names, as well as any input and output program parameters. The flow of control through the modules of the program should also be illustrated.

- ***Appendix II***. A complete source code and any relevant computer output obtained as the result of executing programs developed in the project should be listed here. Hints for printing the source code:

- Print only the code you created. Do not print automatically generated code. If you enhance an old software, do not print it. Do not print code that performs simple functions. If your code contains procedures that are almost identical, print just one.
- Use simple spacing and small font. The source code may be back printed and contain two pages on the side of the sheet.
- Alternatively, prepare a CD with your source code instead of paper copy.

Maximum depth of headers should be 3, i.e. 1.

1.1

1.1.1

It is expected that, excluding the appendices, the project report should not be longer than fifty pages. For guidance with writing style, consult Strunk and White (1979) and Flesch (1981), available in the Thode Library.

4 TYPING AND REPRODUCTION OF THE REPORT

4.1 Typing

The instructions given below apply both to the final report and the Extended Project Proposal. Only *one* copy of EPP is required.

The report must be printed on good quality Letter size paper with a good quality printer. Margins must not be less than 25mm on all sides. Page numbers must be placed in the top, right-hand position. Reports may be sing1e or double spaced, with a font size of 10 to 12.

Diagrams and tables should be drawn in black; do not use colours. Photographs do not reproduce well on photocopying equipment and students should use original photographs. Over-size pages (charts, graphs, tables) may be photo-reduced, but if they are not reduced they should be carefully folded into the report. Figures and tables should have titles and should be listed in a "List of Figures" and a "List of Tables". Ensure that every figure and table can be understood without reference to the text.

4.2 Number of Copies

Each group is to provide two copies of their write-up. The copies must be in large brown envelopes with student names and numbers and group number clearly marked. "Clean" copies are to be submitted – not bound or punched, or stapled.

A *group with departmental supervisor* will provide the supervisor with one copy; the 2nd copy will be submitted to the course coordinator (for an examiner - departmental faculty member - to evaluate it). A *group with an external supervisor and a departmental co-supervisor (liaison)* will provide the both supervisors with a copy (the liaison will be an examiner of the group).

4.3 Deadlines

Some vague deadline here such as ‘beginning’ or ‘middle’ will be specified on our web site under ‘Announcements’. If not, there are interpreted as ‘1st of month’ and ‘15th of month’.

Beginning of Oct: Select project and define problem. Record of Proposed Project Form (see our web site) must be completed and submitted to the course coordinator.

Beginning of Nov: Analyze requirements and design program.

End of Nov: A printed copy of the Extended Project Proposal, approved by your supervisor, must be submitted to the course coordinator. It must include the License to McMaster University (see our web site). The form must be signed by the students and the supervisor. Without this, the group cannot continue and must find another project.
Write code and demonstrate program prototype, primarily showing the user interface, to your supervisor.

Beginning of Feb: Refine and test program.

Beginning of Mar: A draft copy of your report and all supplementary material (including program listing and output) must be submitted to your supervisor(s).

Middle of Mar: Arrange demonstration of finished product with course coordinator, i.e. discuss with him the hardware and software you will need for demonstration.

1.Apr: Make corrections to the report and print final copies of write-up. Project report due.

Note. There will be a late penalty of 10% per day if the above deadlines for EPP and the final write-up are not met.

5 EVALUATION

5.1 Marking Scheme

If you have a departmental supervisor, then you (with the assistance of the course coordinator) will have to find an examiner (a departmental faculty member). If you already have an external supervisor and departmental co-supervisor (liaison), then the liaison will also serve also as the examiner. The examiner will read the write-up and conducts the questioning in the oral examination (See Chapter 5).

The final grade is assigned by the course coordinator using input from the supervisor(s) and examiner, according to the following scheme:

Extended Project Proposal (group mark, marked by the course coordinator)	10%
Written Report (group mark, marked by supervisor and examiner)	30%
Oral Examination (individual mark, marked by supervisor and examiner)	40%
Supervisor(s)' Evaluation (individual mark)	20%

5.2 Evaluation of Project Work

a. *Creativity*: Was the problem solved with originality, resourcefulness, and initiative on the part of the project-group members?

b. *Quality of the Analysis*: Was the scope of the problem understood? Is there a clear statement of what is to be accomplished? Were the tasks sufficiently limited to be achievable, but not trivial? Was there a procedural plan for tackling the problem? Were milestones carefully identified?

c. *Thoroughness*: Was the project completed within the scope of the original objectives; if not, were the reasons clearly given and justified? Were acceptance tests defined and met? Is the documentation adequate?

d. *Skills/Knowledge*: How well were the skills and knowledge taught within the Computer Science programs offered by the department applied to the project? Were any skills and knowledge required other than those taught in the program, and how well did the student develop the new skills or acquire the knowledge?

5.3 Evaluation of the Project Report

a. *Clarity*: Are the purpose, procedure and conclusions presented in a clear and concise manner? Are the program documentation, user guides, etc. readable and understandable at a programmer and user level?

b. *Quality*: Is the report written in good English? Is it free of misprints? Is the quality of print, figures, tables, and flowcharts up to publication standard?

c. *Completeness*: Can the report be read without reference to Appendix I and Appendix II? Is the user guide sufficient for someone unfamiliar with the details of the code?

d. *Attractiveness*: Is the report attractive in appearance? Are the programs laid out in an attractive manner? Is there a good *walk through*, system chart, or summary?

5.4 Oral Examination

These will be held in April, during the University examination period. The specific schedule will be announced after the Registrar's examination schedule is final. The oral examination for a project group proceeds as follows:

1. A half-hour is allotted for the group to demonstrate its project. Students are responsible for arranging availability of hardware and software for the demonstration, which is normally held in ITB. Discussions between students, supervisor, course coordinator, and technical staff may be necessary.

2. The examination continues in the examination room. Each student is given an oral examination of 20 minutes in which the student gives a 10 minute presentation, then the supervisor, examiner, and the course coordinator question the student for 10 minutes. – Only the course coordinator, the supervisor(s), the examiner, and the students of the group are present for the individual examination.

The student's presentation should summarize the project and explain the division of work among the group members. – The oral examination is intended to ascertain the student's understanding of the project and the degree of their participation in it. Usually questioning will be concerned with the design decisions made, the algorithms and techniques used, as well as the explanation of motivation, results achieved, and future extensions that will enhance the product. Questions in the subject area covered by the project are to be expected. Finally, it should be stressed that it is not the intention of the supervisor, examiners, and the course coordinator to trip up the students, but rather to extract their knowledge of the subject area.

BIBLIOGRAPHY

Brooks, Jr., *The Mythical Man-Month: Essays on Software Engineering,* Addison-Wesley Publishing Co., Reading, MA (1995).

Flesch, *How to Write Plain English,* Barnes & Noble Books, Harper & Row Publishers, New York, NY (1981).

Rinearson and J. Woodcock, *Microsoft WORD Style Sheets,* Microsoft Press, Redmond, WA (1987).

Strunk Jr. and E.B. White, *The Elements of Style,* 3rd ed., MacMillan Publishing Co, Inc., New York, NY (1979).

APPENDIX I: Sample Title Page

The next page shows an example of a title page for the project write-up.

APPLICATION OF DIGITAL COMPUTERS TO THE SOLUTION OF VERY DIFFICULT PROBLEMS

by

T. U. Key
B. S. Jones
J. S. Smith

COMPUTER SCIENCE 4ZP6 PROJECT

Supervised by Mr. U. K. Who
A1 Busy Working Company
Last Street
Hamilton, ON
LOL 1T1

and

Dr. How
Department of Computing and Software
(Liaison)

McMaster University
Hamilton, Ontario L8S 4KI
2010

APPENDIX II: License to McMaster University

ALL EXTENDED PROJECT PROPOSALS (EPP) SUBMITTED MUST INCLUDE A COMPLETED LICENSE TO MCMASTER UNIVERSITY. The licence form is available on our web site.