



New Jersey's Science & Technology University

Master's Project Guide

CS/BNFO 700B

CONTENTS

GENERAL PROCEDURES 3
FACULTY RESEARCH INTERESTS 6
PROJECT PROPOSAL FORMAT 7

1. GENERAL PROCEDURES

This section describes general guidelines for the Master's project, including the required proposal, topics, advisors, deadlines, associated courses, etc. Section 2 describes the required format for the project proposal. Section 3 describes the format of the written project report and documentation required for the project.

1.1 Project Proposal

The project proposal is the background and planning document for the project. It must be done with professional care and thoroughness. It must be the product of the student's own original thought and effort. Any use of others' work must be properly referenced in accordance with standards of scholarship and the University Code of Academic Integrity.

The project proposal is written by the student and must be approved by his/her advisor and by the CS department before the student enrolls in the project course CS/BNFO 700B. Master's students should contact a prospective project advisor as early as possible in their course of study since the project advisor may expect certain courses to be taken in preparation for the project. The proposal itself should be developed in consultation with the advisor. It should be initiated by the first half of the semester prior to the semester in which the project is done, and even earlier for some advisors. Completed proposals are submitted to your academic advisor and reviewed for approval by the CS chair.

1.1.1 Proposal Deadlines

For projects that are to begin in a Spring semester, the proposal is due at 2 PM the last day of the eighth week of the fall semester. For projects that are to begin in a Summer term or Fall semester, the proposal is due at 2 PM the last day of the eighth week of the spring semester. If approved, the project is then implemented under the guidance of the advisor.

1.1.2 Submission Procedures

A hard copy of the proposal, with a signed cover sheet as shown in Section 3 of this document, is submitted to the CS department for approval. Submit the proposal on or before 2 PM of the deadline date to your academic advisor. Retain a copy of the proposal for yourself. The intended faculty advisor for the project must be indicated and his/her signature must already be on the cover sheet when the proposal is submitted to the Department.

1.2 Department Action

The CS Department response to the submitted proposal will be emailed. The possible Department responses are: **ACCEPTED**: the student may start work on the project, but should contact their advisor for comments and an advisement schedule; **CONDITIONALLY ACCEPTED**: the student must modify the proposal according to the advisor's indications and approval; upon subsequent approval, the student may then start the project; **REJECTED**: the proposal is unacceptable: the student may contact the faculty reviewer of the proposal for advice.

Proposals that are conditionally accepted must be resubmitted to the Department before the end of the semester.

1.3 Project Topics

Project topics vary greatly. A topic may be suggested by an advisor. A student may propose a topic to the advisor. In any case the student should have some input into the choice of topic. Nevertheless, an advisor has every right to refuse to supervise a project on a student-generated topic. It is important to confirm the appropriateness of a topic with an advisor as early as possible in the development of the proposal.

Master's projects are expected to be conceived, designed, and executed at a graduate level of sophistication. Typically, a project will involve sophisticated software development. Generally, the amount of work required for a project is expected to be comparable to that entailed by one to two graduate courses. Regardless of the type of project, it must be researched in an in-depth and scholarly manner.

1.4 Master's Project Requirements

1.4.1 Master's Project

The course for the Master's Project is CS/BNFO 700B (3 Credits). The section of CS/BNFO 700B that you enroll in depends on your advisor - each advisor has a separate section. You will receive a permit to register by your academic advisor. CS/BNFO 700B has no scheduled hours; that is up to you and your advisor. You must contact your advisor regularly and keep your advisor well informed regarding the progress of your work.

A student who is not able to complete the project in a given semester must notify the advisor one week before the last day of classes and supply a progress report. Normally such students would receive a grade of "S." If your project takes more than 1 semester, then you must enroll in an additional 3 credits for the second semester. Only three credits of a Master's project may be counted towards meeting the MSCS degree requirements. A student who does not enroll in a succeeding semester or who wishes to change advisors must resubmit a proposal prior to the semester in which they plan to do the work. If an advisor change is involved, then both advisors must be aware of and agreeable to the change. Students who find it necessary to use more than 2 semesters to complete the project must request permission to do so from the chairman of the department and provide a written explanation of the circumstances.

1.4.2 Master's Project Reports

Final project reports follow the outline given in Section 3, subject to modifications and/or additions required by your advisor. The report should include CDROM of the source code.

A copy of the final report is submitted to the project advisor for his/her signature. The student submits a CDROM with a signed title and approval page to your academic advisor a week before the first day of finals.

The student cannot graduate unless a copy of the report is given to the department.

Students are advised to have their advisor review a draft of the report well before submission of the final report. It is the student's responsibility to allow sufficient time for the advisor to review and sign the report before it is submitted to the department.

Note: software developed in a Master's project cannot be considered proprietary. The advisor has the right to use such code in research activities and to supply such code to students for use in derivative projects. An advisor may grant a student the right to copyright the code.

1.4.3 Academic Honesty

The project proposal and final report must be the product of the student's own original thought and effort. Any use of others' work must be properly referenced in accordance with standards of scholarship and the University Code of Academic Integrity. Furthermore, a student who, without acknowledgement and the consent of the project advisor, submits a project for which he/she has already received academic credit or monetary compensation may be found guilty of self-plagiarism and subject to penalties under the Honor Code.

2. FACULTY RESEARCH INTERESTS

Go to the Faculty link on the Computer Science home page under People, <http://cs.njit.edu/people/faculty.php> to see what the individual faculty members are researching.

You may not select an adjunct instructor or a special lecturer as your advisor.

Some advisors expect you to take certain courses in preparation for the project.

3. PROJECT PROPOSAL FORMAT

3.1 Introduction

Use the outline on the following pages to prepare your proposal. If you feel you need to deviate from the standard outline, consult with your advisor before doing so; otherwise your proposal will be rejected. Use a word processor to produce the document. Submit the proposal on 8 1/2 by 11 inch paper. Double space all text and use only one side of the paper.

You must staple your proposal in the upper left corner and have your project advisor sign the cover sheet in order to submit the proposal to the department secretary. Do not use the standard NJIT red 3-ring binder for the proposal, but do use it for the final project report.

3.2 Project Proposal Outline

i. *Title Page*: See the sample in Section 3.4 for format.

ii. *Table of Contents*:

1. *Introduction and Background*: The purpose of this section is to describe the general problem area.

1.1 *Problem Statement*: Give a brief, general statement of the problem to be investigated or solved by the project. Assume the reader has little knowledge of the subject.

1.2 *Previous Work*: This is an historical or conceptual survey of relevant work done in the area by previous investigators. Each contribution must be accompanied by appropriate references to be listed in the reference section.

1.3 *Background*: Here you develop the theoretical and conceptual framework upon which the project is based. It is appropriate to describe relevant data representations and algorithms.

1.4 *Glossary*: This section defines all terms, concepts, symbols, and acronyms used in the proposal.

2. *Project Description*: The purpose of this section is to describe the proposed project in detail: what will you do, how will you do it, and when will you do it.

2.1 *Functional Specification*: This is a detailed specification of functions performed by the proposed system, from an external or user perspective, not from an internal or programmer viewpoint. Thus, the system is regarded as a black box with various inputs and outputs related by the functions performed by the system. The description should be sufficient for another programmer to implement the system.

2.1.1 *Functions Performed*: List and briefly describe each of the functions which the system will be designed to perform for its user: What the system will do.

2.1.2 *Limitations and Restrictions*: List and describe each of the internal (self) and external (environment) limitations and/or restrictions on the range of system functions: What will the system *not* do.

2.1.3 *User Interface Design: [if required]* Give a detailed description of the system user interface including diagrams of all the "work" windows (or screens or panes), a table of operations for each work window, and precise descriptions of each operation that the user would regard as unfamiliar. A work window is one that contains data the user is editing, browsing or viewing. This section is required for all programs that engage the user interactively. Refer to the sample in Section 3.4 of this document

2.1.4 *Other User Inputs: [if required]* Give a precise description of the other inputs to the system including source (human or storage) syntax (format) and semantics (meaning). Give examples. This section is required for all programs that obtain input from their environment non interactively

2.1.5 *Other User Outputs: [if required]* Give a precise description of the other outputs of the system including syntax and semantics. Correlate the outputs with the inputs and the functions performed. Give examples. This section is required for all programs that obtain input from their environment non interactively.

2.1.6 *System Data Files;* Give a precise description of the data files created or maintained by the system. Thus, for example, you would include files in a database and you would exclude executable files and text files.

2.2 *Design Specification:* This is a top level *preliminary or provisional* indication of the proposed system architecture and flow. You should correlate system functions with system structure and interface specifications.

2.2.1 *System Data Flow Diagrams:* This is a hierarchical (or leveled) set of diagrams showing the flow of data elements into and out of the functional units of the program, data stores and environmental sources and sinks. Labeled arrows denote data flows. This diagram is complementary to the structure chart described next. Refer to the sample in Section 3.4 of this document.

2.2.2 *System Structure Chart:* This is a (set of) chart(s) showing the functional units of the system hierarchically organized to show which units call, use or contain other units. Each interface between two units (a call) is annotated with small arrows and data item labels to show the data exchanged between the units. Refer to the sample in Section 3.4 of this document.

2.2.3 *System Data Dictionary:* This is a comprehensive dictionary of all the data items that appear in the system data flow diagrams and the structure charts. At a minimum it contains, for each data item, its identifier, any abbreviation used instead of the identifier, the name of the type of the data, and a definition of the data item in the form of either a symbolic expression or a precise description. Refer to the sample in Section 3.4 of this document.

2.2.4 *Equipment Configuration:* Describe the equipment you will use to support the operation and development of your system.

2.2.5 *Implementation Languages:* List the programming languages you plan to use for the implementation of your project and give reasons for choosing each language.

2.3 Implementation Plan: This is a description of the plan for implementing the project. Here you commit yourself to a course of action and specify the criteria by which your performance is to be judged. Your final grade will depend, in large measure, upon your success in achieving the goals agreed upon between you and your project advisor.

2.3.1 Deliverable Items: List and describe each of the items you will submit in fulfillment of the project requirements. Deliverable items include, but are not limited to, program executable file(s), program data file(s), program listings, program documentation, user manual and sample program runs.

2.3.2 Milestone Descriptions: Identify each of the milestones or check points that mark the completion of some phase of project implementation. Milestones include, but are not limited to, detailed system analysis, system design, file design, module design, system test design, module coding, working breadboard with stubs, working system with stubs, system testing and documentation.

2.3.3 Milestone Completion Criteria: List the criteria by which the completion of each milestone is to be judged. If an objective measure is available then it should be specified. If a personal judgment is required then indicate who will make the determination. This information may be given in tabular form if desired.

2.3.4 Schedule of Milestone Completion: Prepare a diagram or table giving the proposed completion date for each of the milestones listed in the previous two sections. See the sample in section 3.4 of this document.

3. References: In this section you list in standard bibliographic format the books, papers, course notes and project or thesis reports which you have used in preparing your project proposal. Each item listed here must be accompanied by a brief comment indicating its importance and relevance to the proposal.

References must be numbered numerically. In the text they should be referred to by squared bracketed numerals, e.g. [1]. The list of references must be typed on a separate sheet(s), in the same format as the main text, and ordered consecutively, according to the following models:

For a paper in a contributed volume:

[1] D.B. Lenat, R. Davis, J. Doyle, M. Genesereth, I. Goldstein and H. Schrobe, "Reasoning about reasoning," in Building Expert Systems (F. Hayes-Roth, D.A. Waterman and D.B. Lenat, eds.) Addison-Wesley Publishing Company, Inc., 1983, pp. 219-240.

For a paper in a journal:

[2] D.W. Jones, "Concurrent operations on Priority Queues," Communications of the ACM, 32, 1(January 1989), pp- 132-137.

For a book:

[3] H.F. Korth and A. Silberschatz, Database System Concepts, McGraw-Hill, New York, 1986.

For an unpublished paper:

[4] S.E. Feahman, A system for representing and using real-world knowledge, MIT Technical Report AI-TR-40, Cambridge, MA, 1977.

4. *Qualifications*: The purpose of this section is to show the advisor that you are capable of successfully completing the proposed project.

4.1 *Personal Background*: Describe any personal experience such as outside employment that is relevant to the proposed project.

4.2 *Courses Taken*: List by department, number and title all courses you have taken. Briefly describe the courses you think have particular relevance to the project.

4.3 *Programs Written*: Describe any programs you have written which either are relevant to the project or are indicative of your ability to complete the proposed work.

4.4 *Investigations*: Describe any work you may have done in preparation for making the proposal such as prototypes or simulations

4.5 *Projects*: Describe any relevant projects you have been involved in or have undertaken.

5. *Grading Criteria*: In this section you establish and define the criteria governing the grading of your project. Here you specify the relative emphasis you wish to be placed upon the different phases of your project. Assign a weight to each of the deliverable items and/or milestones so that the weights sum to one. Display this information in a table which your advisor will use in determining your grade for the project. Refer to the sample in section 3.4 of this document.

3.3 Sample Sections of the Project Proposal

The following page is an example of the title page.

3.3.1 Sample Title Sheet for Proposal

CS 700B Master's Project Proposal
Automated Software Configuration Management
and Change Control System (SCM)

Submitted to the
Department of Computer Science
College of Computing Sciences
New Jersey Institute of Technology

in Partial Fulfillment of
the Requirements for the Degree of
Master of Science
by
John P. McQueue

Student e-mail

APPROVALS

Agree to Advise: _____
(Project Advisor)

Date Submitted: _____

Approved by: _____
(CS Chair)

Date Approved: _____

4. PROJECT FINAL REPORT FORMAT

Check with your advisor for modifications appropriate to your project, particularly for hardware projects. Submit the documentation, one copy to your advisor, and one copy as a CDROM, with the title page and signed approval page accompanying the CDROM, to the department (academic advisor). Please use a jewel case to hold the CD.

Please note the following:

- 1) The report should be printed on one side of the page, double spaced, with wide margins.
- 2) Your report must be complete when you submit it for acceptance.
Pay particular attention to include:
 - Title page
 - Faculty Advisor approval page
 - Contents page
 - Abstract page
 - Text
 - Bibliography
- 3) Number each sheet consecutively at the bottom of the page.
- 4) Headings: Chapter titles start on a new page. Chapter numerals should be Arabic, not Roman numerals. Type the chapter number and title in upper and lower case, flush left, at the top of the report page; leave an extra space and then begin the text. Since you will have several levels of subheadings, distinguish one level from another in a consistent way, such as (1, 1.1, 1.2, 2, 2.1, 2.1.1, 2.1.2, 2.2). Avoid having more than three levels of subheadings.
- 5) Abstract: A brief abstract should be included before the beginning of the text.
- 6) Footnotes should be used sparingly and should be placed at the bottom of the page in which they are referenced.
- 7) All tables and figures should be centered in the column on the paper. Table captions should be centered above the table. All figure captions must appear centered under the figure.
- 8) YOU MUST HAVE THE SIGNATURE OF YOUR PROJECT ADVISOR ON THE FACULTY APPROVAL PAGE.

4.1 Report Outline

Note: Section numbers as shown are mandatory; font, indentation and main section titles in **bold** are merely for clarity here and are optional. This report format can be modified by your advisor. This modification must be explicitly approved by a signed statement at the beginning of the report.

- i Title Page
- ii Acceptance/Approval Page
- iii Abstract
- iv Key Words and Phrases
- v Computing Review Subject Codes
- vi Table of Contents
- 1. Introduction and Background**
 - 1.1 Statement of Problem Area (brief, non-technical)
 - 1.2 Previous and Current Work, Methods and Procedures (representative)
 - 1.3 Background
 - 1.4 Brief Project Description (overview of new, extended or different functions, structure or operation)
 - 1.5 Purpose/Objectives/justification of Project (theoretical, practical, or educational impacts on hardware, software, or users)
- 2. System Functional Specification**
 - 2.1 Functions Performed (itemize and describe)
 - 2.2 User Interface Design [if required]
 - 2.2 Other User Input Preview [if required]
 - 2.3 Other User Output Preview [if required]
 - 2.4 System Data Base/File Structure Preview
 - 2.5 External and Internal Limitations and Restrictions

Prof. Turoff requires the following items as well:

 - 2.6 User Interface Specification
 - 2.6.1 Interface Metaphor Model
 - 2.6.2 User Screens/Dialog
 - 2.6.3 Report Formats/Sample Data
 - 2.6.4 On-line Help Material
 - 2.6.5 Error Conditions and System Messages
 - 2.6.6 Control Functions
- 3. System Performance Requirements**
 - 3.1 Efficiency (speed, size, peripheral device usage)
 - 3.2 Reliability
 - 3.2.1 Description of Reliability Measures (accuracy, precision, consistency, reproducibility, etc.)
 - 3.2.2 Error/Failure Detection and Recovery (failure modes, failure consequences, error logging and reporting, manual and automatic recovery procedures)
 - 3.2.3 Allowable/Acceptable Error/Failure Rate
 - 3.3 Security
 - 3.3.1 Hardware Security
 - 3.3.2 Software Security
 - 3.3.3 Data Security
 - 3.3.4 Execution Security (user validation)

- 3.4 Maintainability
- 3.5 Modifiability
- 3.6 Portability
- 3.7 Others
- 4. System Design Overview**
 - 4.1 System Data Flow Diagrams
 - 4.2 System Structure Charts
 - 4.3 System Data Dictionary
 - 4.4 System Internal Data Structure Preview
 - 4.5 Description of System Operation (high level)
 - 4.6 Equipment Configuration (diagram and description)
 - 4.7 Implementation Languages (which and why)
 - 4.8 Required Support Software (pre-existing)
- 5. System Data Structure Specifications**
 - 5.1 Other User Input Specification
 - 5.1.1 Identification of Input Data
 - 5.1.2 Source of Input Data (NOT input device)
 - 5.1.3 Input Medium and/or Device
 - 5.1.4 Data Format/Syntax
 - 5.1.5 Legal Value Specification
 - 5.1.6 Examples
 - 5.2 Other User Output Specification
 - 5.2.1 Identification of Output Data
 - 5.2.2 Destination of Output Data (NOT output device)
 - 5.2.3 Output Medium and/or Device
 - 5.2.4 Output Format/Syntax
 - 5.2.5 Output Interpretation (meaning of output)
 - 5.2.6 Examples
 - 5.3 System Data Base/File Structure Specification
 - 5.3.1 Identification of Data Base/Files
 - 5.3.2 (Sub)systems Accessing the Data Base (creating, updating, using; frequency)
 - 5.3.3 Logical File Structure (record formats, file organization, access methods, rationale, examples)
 - 5.3.4 Physical File Structure (storage device, blocking, organization, access, etc.)
 - 5.3.5 Data Base Management Subsystems Used (internal or external)
 - 5.3.6 Data Base Creation and Update Procedure (if NOT by system)
 - 5.4 System Internal Data Structure Specification
 - 5.4.1 Identification of Data Structures
 - 5.4.2 Modules Accessing Structures (creating, updating, using)
 - 5.4.3 Logical Structure of Data (format, organization, access, rationale, examples)
- 6. Module Design specifications (for each module)**
 - 6.1 Module Functional specification
 - 6.1.1 Functions Performed
 - 6.1.2 Module Interface Specifications (input/output arguments/global variables/files)
 - 6.1.3 Module Limitations and Restrictions
 - 6.2 Module operational Specification
 - 6.2.1 Locally Declared Data Specifications (variable dictionary)
 - 6.2.2 Algorithm Specification (flowchart, pseudo code, decision table, etc)

6.2.3 Description of Module Operation

7. **System Verification**

7.1 Items/Functions to be Tested

7.2 Description of Test Cases

7.3 Justification of Test Cases

7.4 Test Run Procedures and Results

7.5 Discussion of Test Results

Prof. Turoff requires the following items as well:

7.6 Evaluation of User System

7.6.1 Protocol Study

7.6.2 User Survey

7.6.3 Real Time Monitoring

7.6.4 Interviews

8. **Conclusions**

8.1 Summary

8.2 Problems Encountered and Solved

8.3 Suggestions for Better Approaches to Problem/Project

8.4 Suggestions for Future Extensions to Project

9. **Bibliography**

10. **Appendices**

11. **Program Listings**

12. **User Manual**

4.2 Sample pages

4.2.1 Title Page

**Automated Software Configuration Management
and Change Control System (SCM)**

Submitted to the
Department of Computer Science
College of Computing Sciences
New Jersey Institute of Technology

in Partial Fulfillment of
the Requirements for the Degree of
Master of Science
by
John P. McQueue

4.2.2 Approval page

APPROVALS

Approved by: _____
(Project Advisor)

Date Submitted: _____