



UNIVERSITY OF EDUCATION

Final Project Proposal Guide

TABLE OF CONTENTS

FINAL PROJECT PROPOSAL GUIDE	3
1. INTRODUCTION	3
1.1 PROJECT TITLE	3
1.2 PROJECT OVERVIEW STATEMENT	4
1.4 PROJECT GOALS & OBJECTIVES.....	6
1.5 HIGH-LEVEL SYSTEM COMPONENTS.....	6
1.6 LIST OF OPTIONAL FUNCTIONAL UNITS	6
1.7 EXCLUSIONS	7
1.8 APPLICATION ARCHITECTURE	7
1.9 GANTT CHART	7
1.10 HARDWARE AND SOFTWARE SPECIFICATION	8
1.11 TOOLS AND TECHNOLOGIES USED WITH REASONING	8

Final Project Proposal Guide

1. Introduction

This guide will tell you how to prepare and submit the final project proposal that is the documented work for the Project. A good project proposal must define the functional and non-functional requirements in unambiguous statements, Scope of the Project, Development Schedule, Development Process, Techniques, Tools, Platform with reasoning. However, a professional and well-defined proposal should be composed under the following headings;

- a. Project Title
- b. Project Overview Statement
- c. Project Goals
- d. Project Objectives
- e. High Level System Components
 - a. Component no.1
 - b. Component no.2
 - c. Component no.3
 - d. Component no.4
 - i.
- f. List of optional functional units
- g. Exclusions
- h. Application Architecture
- i. Gantt chart

1.1 Project Title

The title should be clear and unambiguous (do not make it "cute"). Think of your title as a mini-abstract. A good title should paint a quick picture for the reader of the key idea(s) of your project. The words you use in your title should clearly reflect the focus of your proposal. The most important words should come first, then the less important words. Try to remove words from your title that really are not necessary for understanding. Try and use only a single sentence for your title. If the sentence is getting too long, try removing some words. When all else fails try using a two-part title with the parts separated by a colon (use only as a last resort!). Do not attempt to use the title as an abstract of your entire proposal. If your proposal is built on collaborating with other groups/organizations, it is usually a good idea to include their names on the Title/Cover Page.

1.2 Project Overview Statement

Think of the Project Overview as an Executive Summary (the busy executive probably only has enough time to read your Overview - not the entire proposal). Be specific and concise. Do not go into detail on aspects of your proposal that are further clarified at a later point in your proposal. The Project Overview should "paint a picture" of your proposal in the mind of the reader. It should establish the framework so that the rest of the proposal has a frame of reference. Use the Project Overview to begin to show your knowledge of the organization from which you are requesting funds. Key concerns of the funding organization can be briefly identified in relation to your proposed project. If you will be collaborating with other organizations, make sure some of their interests are also highlighted in the Project Overview. This can assist in strengthening the collaboration by recognizing them at the very beginning of your proposal. The best time to prepare the Project Overview is after you have completed the entire proposal (and you understand all aspects of your proposal very well). Let the Overview be your last piece of writing and then insert it at the beginning of your proposal. Try to keep in mind that someone will be reviewing your proposal and you would like to have this person be very positive about what you have written. The Project Overview will probably form a strong impression in the mind of the reviewer. Work on your Project Overview so that you can avoid giving this person the opportunity to say things like:

1. Not an original idea
2. Rationale is weak
3. Writing is vague
4. Uncertain outcomes
5. Does not have relevant experience
6. Problem is not important
7. Proposal is unfocused
8. Project is too large.

Project Overview Statement Template

Project Title:			
Group Leader:			
Project Members:			
Name	Registration #	Email Address	Signature
Project Goal:			
Objectives:			
Sr.#			
1			
2			
3			
4			
5			
6			
Project Success criteria:			
Assumptions, Risks and Obstacles:			
Organization Address (if any):			
Type of project: <input type="checkbox"/> Research <input type="checkbox"/> Development			
Target End users:			
Development Technology: <input type="checkbox"/> Object Oriented <input type="checkbox"/> Structured			
Platform: <input type="checkbox"/> Web based <input type="checkbox"/> Distributed			
<input type="checkbox"/> Desktop based <input type="checkbox"/> Setup Configurations			
<input type="checkbox"/> Other			
Suggested Project Supervisor:			
Approved By:			
Date:			

1.4 Project Goals & Objectives

Try and differentiate between your goals and your objectives - and include both. Goals are the large statements of what you hope to accomplish but usually aren't very measurable. They create the setting for what you are proposing. Objectives are operational, describe specific things you will be accomplishing in your project, and are very measurable. Your objectives will form the basis for the activities of your project and will also serve as the basis for the evaluation of your project. Try to insure that there is considerable overlap between the goals and objectives for your proposal and the goals and objectives of the funding organization. If there is not a strong overlap of goals and objectives, then it might be best to identify a different funding organization. Measurable objectives for your project should be presented. If you are dealing with "things" it is easier for them to be measured than if you are dealing with abstract ideas. Your proposal is easier for a prospective funding organization to understand (and the outcomes are much more clear) if you describe your objectives in measurable ways. The following are the objectives of project:

- a- Prepare List of all objectives.
- b- -----
- c- -----

1.5 High-level system components

Information about the main functional units of the entire system should be present. Functional units to be included will be the inclusive components of the project developed so that the system must perform without taking any physical constraint into consideration. High-level system components are generally, a set of cooperating components assembled together to deliver a solution to a problem. They are frequently identified in terms of inputs, outputs, processes, and stored data that are needed to satisfy the system improvement objectives. If these components are missing the system fails to fulfill its primary mission.

1.6 List of optional functional units

A list of functional units should be present which would include a description of other features, characteristics, and constraints that define a satisfactory system. These functional units would be developed under certain conditions (technology, expertise, or time dependent). Examples of these optional functional units would include performance (throughput and response time); ease of learning and use; budgets, costs, and cost savings; timetables and deadline; documentation and training needs; quality management; and security and internal auditing controls.

They are often requirements that specify need of compliance with any legal and regulatory requirements. They can also be design constraints due to the operating system used, the platform environment, compatibility issues, or any application standards that apply. In general, you can say that any requirement that does not allow for more than one design option should be regarded as a design constraint.

If the optional functional units are missing the system can still (for a while) fulfill its fundamental mission, but with degraded service quality.

While gathering and validating the optional functional requirements, maintain Assumptions and Issues lists.

Some activities will not give you satisfactory answers. This can be due to lack of information, or simply because you consider the answer threatens the viability of the design. Therefore, create two lists, and maintain them through the design study:

Any assumptions you make during the requirements and design process, including the rationale or thought processes behind those assumptions. Assumptions may be used to identify related subprojects or items of work, which are outside the scope of or after this project any major issues (significant concerns that could become show-stoppers).

The issues should be reviewed with the customer at the end of each phase. The assumptions need to be reviewed also, at the end of each phase, but the customer might not always be the correct person for the less important ones. Assumptions and issues apply to all artifacts, but are particularly common for non-functional requirement.

1.7 Exclusions

A list of the functional units, which will not be intended to be develop or discussed during any point in the project development, should be present. Time constraints or lack of resources for the fulfillment of the required task or any sort of other constraint preventing the completion of the functional unit could be described here.

1.8 Application Architecture

Defines the overall application architecture e.g. a two-tier architecture or a three-tier architecture. It must contain a diagram depicting the system architecture properly Architecture is the highest-level concept of a system in its environment. The architecture of a software system (at a given point in time) is its organization or structure of significant components interacting through interfaces, those components being composed of successively smaller components and interfaces.

Architecture can also be defined as the organizational structure of a system. Architecture can be recursively decomposed into parts that interact through interfaces, relationships that connect parts, and constraints for assembling parts. Parts that interact through interfaces include classes, components and subsystems.

There are a number of typical patterns of distribution in systems, depending on the functionality of the system and the type of application. In many cases, the distribution pattern is informally used to describe the 'architecture' of the system, though the full architecture encompasses this but also many more things. For example, many times a system will be described as having 'client-server architecture', although this is only the distribution aspect of the architecture.




1.9 Gantt chart

The Gantt chart enumerates the activities to be performed on the vertical axis and their corresponding duration on the horizontal axis. It is possible to schedule activities by

either early start or late start logic. In the early start approach; each activity is initiated as early as possible without violating the precedence relations. In the late start approach; each activity is delayed as much as possible as long as the earliest finish time of the project is not compromised.

Based on the Work Breakdown Structure (WBS), a timeline or Gantt chart showing the allocation of time to the project phases or iterations should be developed. This Gantt chart would identify major milestones with their achievement criteria. It must contain duration estimation of all the necessary activities to be carried out during the project development along with the human resources responsible for the respective tasks. Activity dependencies are also required to be mentioned in it.

Sample Gantt chart

ID		Task Name	Duration	Start	Finish	Predecessors	Jul 13, '03							Jul 20, '03								
							W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T
1																						
2		billing	7 days	Thu 7/10/03	Fri 7/18/03																	
3		computing	8 days	Mon 7/14/03	Wed 7/23/03																	
4		accounting	3 days	Mon 7/14/03	Wed 7/16/03																	
5		marketing	10 days	Mon 7/21/03	Fri 8/1/03	2																

1.10 Hardware and Software Specification

Any hardware or software specifications e.g. machine type required, operating system and other utilities should be clearly specified for the system to be developed.

1.11 Tools and technologies used with reasoning

The application tools, which are to be used on front and back end of the system to be developed, should be listed. The reasons for these tools should also be enlisted.

Identify what the needs for tool support are, and what the constraints are, by looking at the following:

Designing Tools:

- Rational Rose for UML diagrams can be one example.
- The development process. What tool support is required to effectively work? For example, if the organization decide to employ an iterative development process, it is necessary to automate the tests, since you will be testing several times during the project.
- Host (or development) platform(s).
- Target platform(s).
- The programming language(s) to be used.

Development Tools:

- Eclipse for java
- Existing tools. Evaluate any existing and proven tools and decide whether they can continue to be used.

- The distribution of the development organization. Is the organization physically distributed? Development tools generally support a physically distributed organization differently.
- The size of the development effort. Tools support large organizations more or less well.
- Budget and time constraints

References:

[1]