Guide to Preparing the
PROJECT CHARTER DOCUMENT

© R. Buckley
CSc 190 Senior Project
Department of Computer Science - College of Engineering and Computer Science
California State University, Sacramento
Version 2.25.2014

The design and layout of each of the CSc 190 and CSc 191 documents should be the same.

Title Page. The document's title page should include, at a minimum, the document title, the project name and the team's name.

Table of Contents. The table of contents should follow the title page. The numbers, titles and page numbers for each section and first level subsections, as well as the letters and titles for each appendix should be included. Second and lower level subsections may be included when reference is needed to provide the reader with direct reference to significant content.

Document Footer. The footer specification for each document should include the document title and page number.

Readability. The team's writing, reviewing and revision process should ensure that the information contained in each document will be clear and understandable to its intended audience.

This team's final draft of each document must then be submitted to the team's faculty adviser for approval. Faculty adviser approval requires that the quality and readability of each document satisfy the standards required of all computer science majors. In addition, the Charter, Software Requirements, and User Manual documents must be formally approved by the project's sponsor.

Upon completion of the project, all the documents will be delivered to the sponsor. The information contained in the Software Requirements Specification, the Software Design Specification and the System Test Specification documents should facilitate whatever updates and/or maintenance of the software will be needed after the project team delivery

The Table of Contents should follow the title page and the INTRODUCTION (Section 1) should begin on a new page. The following example uses this Guide to construct the Table of Contents.

Guide to Preparing the
PROJECT CHARTER DOCUMENT
Table of Contents

The Table of Contents should follow the title page and the INTRODUCTION (Section 1) should begin on a new page.

1 INTRODUCTION

Section 1 serves as the Executive Summary for the document. It introduces the reader to the contents of document.

What follows is sample wording that can be used in this section:

> This is the Project Charter document for the <*name of the project*> project sponsored by <*name of sponsor*>.
>
> This project is being undertaken by the <*name of team*> development team. The team is comprised of undergraduate students majoring in Computer Science at California State University, Sacramento. The team members are enrolled in a two-semester senior project course required of all undergraduate majors. Successful delivery of the desired software product will fulfill the senior project requirement for the student team members.
>
> *Include the identification of the project sponsor and the team as follows:*
>
> > PROJECT SPONSOR (*if there is more than one person serving as sponsor, list each sponsor*):
> > Name
> > Title
> > Company or Organization name
> > Contact information (phone number and Email address)
> >
> > <*team name*> DEVELOPMENT TEAM
> > List of team member names
> > Team contact information (phone number and Email address)

The remaining subsections in Section 1.0 are intended to introduce the reader to the document and to serve as a high level executive level summary of its contents. Write this section last! At the outset the project team may have a general understanding of the purpose and scope of the

document but one that is not sufficient to inform the general reader. In most cases, it is best to leave this section to the very end, or at least reserve final review and revision until a draft of the document has been completed.

1.1. Purpose. This subsection should describe the purpose of this document. Explain why this document is being written, what it is intended to communicate, and what the reader should expect, in general, to learn. The purpose statement should capture the intent associated with the contents of the document. In the case of this document, your intent is to create a mutual understanding between the team and the sponsor of what is expected over the course of the project.

Note: The Project Charter is a relatively brief document. It is not a plan and therefore does not provide details as to how the project work will be scheduled over the two semester "life" of the project. In addition, it does not explain how the team will manage the project. It should, however, contain enough information for the sponsor and the team to get started. /The project management plan and the team's planning process will be provided in the Software Project Manageement Plan (SPMP).

1.2. Scope. This subsection should describe what specifically is to be covered by the document and what is not. The reader should be informed as to the context of the information being provided in the document. For example, the contents of this document should not be interpreted as containing a complete set of agreed upon requirements.

1.3. Definitions, Acronyms and Abbreviations. This subsection serves as a glossary for the document. The glossary should help the development team understand the terms used by the sponsor and the user community. The glossary should also help the sponsor and user community understand the technical terms used by the development team. All terms used as well as all acronyms and abbreviations used should be arranged in alphabetical order.

Again, the purpose of this subsection is to provide the reader with quick access to the references used throughout the document. Separate the definitions, acronyms and abbreviation into the following three subsections.

1.3.1. Definitions.

For example:

> Baseline. A baseline is a work product that has been formally reviewed and accepted by the involved parties. A baseline is changed only through formal configuration management procedures.
>
> Software Requirements Specification. Documentation of the essential requirements (functions/features/uses, performance, design constraints, and attributes) of the software and its external interfaces.
>
> Protocol. A set of conventions that govern the interaction of processes, devices, and other components within a system.
>
> Task. The smallest unit of work subject to management accountability.

1.3.2. Acronyms (definition and/or explanations may need to be included for those acronyms that are not commonly used or understood)

List of acronyms

1.3.3. Abbreviations (again, definition and/or explanations may need to be included for those acronyms that are not commonly used or understood)

For example:

MGT. Management
LRN. Learning

Note: The following is the general rule for the use of acronyms and/or abbreviations in a technical document. For the first appearance/use in the document, the acronym or abbreviation should be spelled-out followed by the acronym or abbreviation in parentheses. The acronym or abbreviation is used in all subsequent references.

1.4. References. This subsection serves the same purpose as a bibliography. If any documents were used in the preparation of the document, the formal bibliographic information should be specified here. A reference to a specific bibliographic entry should be made at the location in the document where information from that specific source is used. Included here should be all documents provided by the sponsor.

Use the Modern Language Association (MLA) citation rules and conventions for acknowledging sources used in a preparing the document. One of the many Web links that contains MLA citation examples is:

http://www.library.cornell.edu/newhelp/res_strategy/citing/mla.html

Examples:

Johanna Rothman. *Manage It! Your Guide to Modern, Pragmatic Project Management*, The Pragmatic Programmers, LLC, Raleigh, NC, and Dallas, TX, 2007.

Karl Weigers, *Software Requirement: Practical techniques for gathering and managing requirements throughout the product development cycle*, Microsoft Press, Redmond, Washington, 2003.

1.5. Overview of Contents of Document. This subsection briefly describes each of the remaining sections in the document as well as the contents of each appendix. You will have a subsection for each additional section (beginning with Section 2) in the document. Each subsection should use the exact wording that introduces each of the sections and each appendix in the document.

2 PROJECT SPONSOR AND SPONSOR NEED

This section identifies the project sponsor and describes the sponsor's "business". After reading the following subsections the sponsor should be convinced that the team understands the sponsor's organizational structure as well as its business. An understanding of the business is necessary to for the team to understand the context in which the proposed software will be used.

In addition, the section clearly describes the value the sponsor expects that the software will provide. This description should be the result of the team's initial meetings with the sponsor. Also included, if appropriate, should be a description of the sponsor's current method for satisfying this need.

2.1 Sponsor Identification. This subsection should clearly identify your senior project sponsor. Include the complete organization name (for example, company, division and department names) along with the individual names and titles of your primary contact as well as any others from the sponsoring organization that you will be involved with during the course of the project.

2.2 Sponsor's "Business". This subsection should present the project team's understanding of the "business" in which the proposed software product will be used. The intent is to demonstrate to your sponsor that you have a clear understanding of the sponsor's business, how it is organized, and how it goes about providing goods and/or services. Such understanding should provide the team with an understanding of the context in which the proposed software will be used.

2.3 Description of the Need. This subsection should describe, in general, what the project needs to accomplish and whatever constraints on the project are indicated by the sponsor. The subsection includes a statement of the project vision, which represents the sponsor's reason for the project, the goals that the team hopes to accomplish over the course of the project and what should be considered as the success criteria for the project.

Your explanation should identify the key features to be provided by the software in sufficient detail to convince the sponsor that the project team understands the full nature of the specified need and the context in which it exists.

In most cases the sponsor's current processing approach is the source of the need. If so, describe this process. Include the resources used and the problems that are associated with the current method.

A common fault in writing a Project Charter is to concentrate on the details of what you, the project team, want to code rather than on what is really required. Be careful, for this document is the first formal communication with your sponsoring organization and can affect the way in which the team is perceived over the life of the project. A negative first impression is not easily changed.

Note. If the proposed project is described as a response to a problem, then that response should be described as a solution. If the project is described as a response to a need, then the response should be described as meeting or satisfying that need. Do not mix these metaphors.

2.3.1 Vision. Specify the sponsor's primary reason(s) for proposing the project. Indicate what value will be provided by the project. This vision statement should be convincing and sufficient to motivate the team. This is not a statement of a listing of desired functionality or even a listing of features.

An approach suggested by Joanna Rothman is to identify the sponsor's expectations, that is, what values he or she expects the software provide. The overarching (most important) expectation should provide the team with clear focus on the purpose of work over the course of the project. The team needs to be clear on what appears to be the key expectation associated with defining the project's success. This one item would then be the team's *driver*.

Of the remaining set of expectations, the team needs to identify two or three that will serve as constraints (i.e. constraints in the sense that the team remains focused on what should "drive" the project but understands the constraints on their work). What expectations are left? These are in the nice-to-have category, meaning that they may be the subject of tradeoffs that may be needed to be resolved later in the project.

The point of this activity is to clearly define what it is that the team is to focus on.

2.3.2 Goals. This subsection indicates that the senior project experience is designed to accomplish two goals:

1) To develop and deliver a software system to the benefit of the project sponsor.
2) To provide the senior project team with a learning experience in which software engineering principles are applied to the development of a user specified software system.

In addition, a statement or statements should be included which indicates what the team and its members hope to accomplish by undertaking the project. This would include team members' professional goals. Success in achieving these goals would be the consequence of the process and experience provided through the development of the software system to be delivered to the sponsor.

2.3.3 Success Criteria. This subsection should include a statement of what the sponsor hopes to be able to do with the software once it is delivered. This should be a *general* statement of the product's key and essential capabilities, its key features.

Note. Each feature will require functionality needed to provide the user with the intended purpose of the feature. By first identifying the key features, the team can then begin to question and elicit the functionality that will be needed to design and implement each of the features. Detailed specifications for these functions will be elicited, analyzed and incorporated into the project's software requirements specifications.

Example: A university's registration system would have a "student registers for classes" feature. This task requires quite a bit of functionality as well as access to a database. An understanding of this feature would require an understanding of the user's actual task - registering for classes. This would be a prerequisite to the "interaction design" necessary to facilitate the user's task. This functionality is required to implement the interaction design associated with each feature.

Programmers begin by looking for functions to code. Interaction designers are looking for the best way to deliver this needed functionality through the design of the software's interaction with the user. Understanding of how the user can best achieve his or her objective comes first. Identification of the needed functionality follows. Features are what your program can do. Features are a direct result of user requirements, and business objectives. Thus a program's features exist mainly to meet user demands and sattisfy the sponsor's expectations.

Functionality, on the other hand, is how the aforementioned features are actually implemented.

3 MANAGEMENT PROPOSAL

This section should specify, in general terms, how the project team intends to manage and control the project development.

3.1 Work Schedule. This section should identify the phases of work in the project's software development life cycle that will be completed each semester. For each phase, briefly describe the work associated and its estimated start and completion dates. Completion of each phase culminates with the completion and approval of a baseline product. The final product is the delivery and installation of the software.

Over the course of the project, much of the work is to be documented and approved by the team and the team's faculty adviser. While the Project Charter must be read and approved by the sponsor, not all of the published work is required to be approved by the sponsor. However, the sponsor may request copies for review. A list of the documents and indication of those requiring sponsor review and approval should be provided in the Appendix to this document (and referred to in the body of this document).

3.2 Resource Requirements. The project team should specify an initial estimate of person-hours and support resources (e.g. copying costs and supplies) needed for a typical senior project.

Included should be an agreed upon schedule of meetings between the team and the sponsor over the life of the project. A typical team and sponsor commitment *requires* at least two meetings each month with the expectation that additional meeting would be scheduled as needed. The project team will be responsible for preparing the agendas in advance of each meetings as well as publishing the minutes following each meeting.

3.3 Cost. Typically, this subsection would take the tasks, the resource requirements, and the schedule and estimate the cost a typical senior project.

The following paragraph should be included in this section of the document.

> Senior projects, while "expensive" in the use of team members' time, are undertaken with no expected cost to the sponsor. Consequently, the costs estimated in this subsection are hypothetical and have been developed as examples to illustrate cost estimation concepts used in proposal writing. As such, the cost estimates do not represent costs expected to be incurred or reimbursed.

3.4 Organization and Staff. This subsection should identify the members of the project team and the roles that they will perform over the life of the project (in both CSc 190 and CSc 191). The

team should rotate some of the key responsibilities thereby providing each member with a richer set of experiences in the development process. In addition, one of the team members must be designated to serve as the project manager for CSc 190 work and another team member for the CSc 191 work.

3.5 Quality Assurance. Mention should be made as to how the team will maintain some degree of quality control during each of the phases. You can use this section to describe the procedures the team will use to evaluate the effectiveness and efficiency of their work over the course of the project. Such procedures may consist of regularly conducted technical reviews which may include the project team's faculty adviser, as well as involvement of the sponsor in the approval of baseline products.

3.6 Change Control Process. Changes to prior agreements can be expected during the development period. This typically includes requested changes and/or additions to the project requirements that may involve both the features and the functionality needed to support the features.

A Change Control Process should be agreed to and clearly specified in this document. The process would include the assessment of all requested changes to determine the potential impact on the scope of work and the ability of the team to deliver the project on time. If the assessment indicates that the increase in the scope of work will impact delivery, the team and the sponsor should negotiate and agree to changes that insure that the highest priority features and/or functions are implemented and delivered.

4 CONDITIONS AND COMMENTS

This section contains two subsections that specify the terms and conditions of work by the team and the needed collaboration between the team and sponsor.

4.1 Assumptions and Constraints. This subsection should contain a detailed list of all assumptions being made by the project team as well as constraints placed upon the team by the project sponsor. Answers to the following questions may help in fully assessing the nature and extent of any such assumptions and constraints.

1) What factors exist which provide a fixed limit to the project? Clearly, there are time constraints and technical constraints that exist due to the academic nature of the project.
2) Does the sponsor assume any such constraints? If so, are they consistent with the requirements of senior project?
3) What assumptions the project team is making?
4) Is the sponsor assuming that the system being developed will be compatible with other systems?
5) Is there an assumption that additional documentation will be required in order for the developed system to operationally useful?
6) Are there any critical external activities and/or events that must occur during the development process?
7) Will the project team require timely delivery of hardware and/or software in order to maintain schedule and delivery?

4.2 Limiting Conditions. This subsection should describe whatever limits the project team is placing on the project and the finished product. For example, if the proposed solution will not completely satisfy the sponsor's needs, what specifically is to be included in the software and what is not? The team should clearly identify what they are assuming responsibility to develop and deliver. In so doing, this subsection should clearly identify what would constitute a successfully complete project, that is, what the sponsor should expect prior to the final approval and acceptance of the software product.

This subsection should also contain the team's expectations regarding the need to agree on the importance of sharing the development responsibilities with the sponsor. The sponsor and the team should clearly specify the time commitment necessary to ensure that the development effort will be successful.

There are shared responsibilities that both the team and sponsor should agree upon. Appendix B explains the basis for viewing the development work as a partnership of shared responsibilities. The lists provided can be used to communicate to the sponsor the specific nature of these responsibilities and expectations. Both the team and the sponsor should review and agree to both rights and responsibilities expected of the sponsor.

4.2.1 Factors Associated With the Academic Nature of the Project. This subsection contains a variety of issues that need to be documented because of the quasi-academic nature of the work done by the project team.

4.2.1.1 General Disclaimer. What follows is a sample of the words the team can adapt for use in this section.

> All students majoring in Computer Science at CSUS are required to complete a two semester, senior project. The project proposed, <*project-name*>, is expected to fulfill this requirement for the project team of <*names*>. While the intent of the team is to deliver a high quality product that meets the sponsor's expectations, neither the students, faculty adviser, or CSUS can be held responsible for any errors in the delivered software product, failure to meet any of the specified requirements, or failure to deliver the software.
>
> Furthermore, due to the academic nature of the experience and its requirement for graduation, students can not be paid for the work associated with the project.

4.2.1.2 Support Limitations. The team should also indicate that by accepting this proposal, the sponsor recognizes that upon completion of the project and delivery of the proposed system, neither the senior project team nor any other representative of CSUS is obligated to provide software maintenance or additional support. Senior project work cannot be extended beyond the completion date set for CSc 191.

4.2.1.3 Ownership of the Product. Typically, there are no formal agreements as to the ownership of the software. Senior project is an academic requirement and is not intended to be considered as work by the University or the project's sponsor in which some form of remuneration is expected.

While the software and all of the supporting materials are delivered to the sponsor as a condition of completion of the project, team members share ownership with the sponsor. If the sponsor

requires clear legal title to the software (or some other arrangement), a separate agreement should be prepared and signed by the sponsor and team and included in the appendix to this document.

If no special arrangements are necessary, merely include in this subsection that the team members maintain nominal ownership and the sponsor will receive all the specified documentation along with the software, including both source and executable code. Also, include the stipulation that the Computer Science Department reserves the right to use the documentation and product as examples of student work.

4.2.2 Other Disclaimers. The team should include in this section any additional conditions that the project team requires and/or that require the sponsor's approval.

5 APPROVALS

The introduction to this section should indicate what specifically is being agreed to. In this case, the following sign-off sheet should be used to indicate approval of and agreement to the conditions and commitments contained in the Project Charter.

The signatories should include the project sponsor, each member of the project team, and the project's faculty adviser. The sign-off sheet should include the name, full title, and full name of the parent organization for each of these signatories.

This signed Charter serves as a de facto "contract" between the project team (the developer) and the customer.

APPENDICES The content of appendices contains information that is (and must be) referenced in the body of the document.

APPENDIX A. Project Team Experience.

This appendix should provide information about the qualifications of each member of the development team. The resumes used here should be those prepared for the team's prospectus.

APPENDIX B. Partnership requirements between the team and the sponsor. Karl E. Wiegers authored the following two lists that speak to the need for both the team and the sponsor to share the responsibility of ensuring the software product that is developed is based on accurate and complete requirements. The lists along with an explanation of the items in each of the lists are available at the following web address; http://www.processimpact.com/articles/customer.pdf.

As Wiegers suggests, the sponsor and the team should review the items in the list, adding and modifying as appropriate. In so doing, both parties should reach an understanding regarding this shared responsibility.

B.1 Requirements Bill of Rights for Software Customers

The project sponsor has the right to:

1. Expect the team to speak your language.

2. Expect the team to learn about your business and your objectives for the system.
3. Expect the team to structure the requirements information you present into a software requirements specification.
4. Have the team explain requirements work products.
5. Expect the team to treat you with respect and to maintain a collaborative and professional attitude.
6. Have the team present ideas and alternatives both for your requirements and for implementation.
7. Describe characteristics that will make the product easy and enjoyable to use.
8. Be presented with opportunities to adjust your requirements to permit reuse of existing software components.
9. Be given good-faith estimates of the costs, impacts, and trade-offs when you request a requirement change.
10. Receive a system that meets your functional and quality needs, to the extent that those needs have been communicated to the team and agreed upon.

B.2 Requirements Bill of Responsibilities for Software Customers

The project sponsor has the responsibility to:

1. Educate team about your business and define jargon.
2. Spend the time to provide requirements, clarify them, and iteratively flesh them out.
3. Be specific and precise about the system's requirements.
4. Make timely decisions about requirements when requested to do so.
5. Respect developers' assessments of *scope* and feasibility.
6. Set priorities for individual requirements, system features, or use cases.
7. Review requirements documents and prototypes.
8. Promptly communicate changes to the product's requirements.
9. Follow the team's defined *requirements change process*.
10. Respect the requirements engineering processes the team uses.

A good reference on the entire process of specifying requirements is Karl Weigers 2003 text, Software Requirements: Practical techniques for gathering and managing throughout the product development cycle from Microsoft Press. (ISBN: 0-7356-1879-8).

APPENDIX C. This appendix identifies the phases of work in both CSc 190 and CSc 191, the deliverables associated with each phase, and the approval requirements.

The following table identifies each of the phases of work that are to be completed in developing the software for the team's project sponsor. Associated with each phase is a baseline deliverable indicated in the second column. The last column indicates those deliverables that must be reviewed and approved by the sponsor. The team's faculty adviser is responsible for reviewing and approving the final draft of each baseline document. In the case where the sponsor must also approve the document, the faculty adviser's approval is necessary before submitting the document to the sponsor.

PROJECT PHASE	**PHASE DELIVERABLE**	**APPROVALS ***
Establish the Vision and Scope of the Project	Project Charter	Sponsor

PROJECT PHASE	PHASE DELIVERABLE	APPROVALS *
Develop and define the project management plan	Project Management Plan	
Elicit, analyze, analyze, specify, validate, and publish the requirement specifications	Software Requirements Specification	Sponsor
Design the software	Software Design Specification	
Implement the software design specifications	The Software	
Develop and define the system test plan and specify all necessary test cases	System Test Plan and Test Cases	
Perform system testing and publish the results	Testing & Software Test Report	
Prepare materials to be delivered to the sponsor at the final product acceptance meeting	Software Delivery Materials (includes the User Manual and Delivery CD)	Sponsor
Time spent developing the skills and knowledge necessary to complete the project	Learning (all phases)	NA
Time spent by the team and its members in the management and control of the project	Project Management (entire project)	NA

- All final documents must be approved by the team's faculty adviser. For those requiring approval by the sponsor, the project adviser must first approve the document before it is provided to the sponsor.

The standard for expected hours of out of class course work for each student is a minimum of three (3) hours per week. Both CSc 190 and CSc 191 are two unit courses, consequently this standard translates to a minimum of out of class work time per team member of six hours per week. For each fifteen week course, each team member is expected to work a minimum of 90 hours. Although there is no final in either course, the hours of work that a student would be expected to accumulate in preparation for both midterms and final exams should be added to this minimum work requirement.

Each senior project varies in its scope and complexity and therefore in the time required of the development team. If each team member spends a minimum of 200 hours working on the project (100 hours for CSc 190 and CSc 191), a five member team would spend a minimum of 1,000 hours. The following table lists the estimated percentage of work associated with each phase. These percentages represent averages from previous projects.

PROJECT PHASE	EST % OF WORK
Establish the Vision and Scope of the Project	4%
Develop and define the project management plan	4%
Elicit, analyze, verify and publish the requirement specifications	14%
Design the software	9%
Implement the software design specifications	21%
Develop and define the system test plan and specify all necessary test cases	7%
Perform system testing and publish the results	2%

PROJECT PHASE	EST % OF WORK
Prepare materials to be delivered to the sponsor at the final product acceptance meeting	2%
Time spent developing the skills and knowledge necessary to complete the project	13%
Time spent by the team and its members in the management and control of the project	24%
TOTAL	100%