

GUIDELINES ON WRITING A GRADUATE PROJECT THESIS

SHAN BARKATAKI, COMPUTER SCIENCE DEPARTMENT, CSUN

1. PURPOSE AND INTRODUCTION

The purpose of this document is to provide guidelines on writing a graduate project thesis. It is not intended to be used in writing a thesis describing theoretical research work.

A graduate project thesis represents the culminating experience resulting from your graduate study. Your thesis is the most important artifact you create in earning your degree. It will persist in perpetuity, long after your graduation. It is the thesis that proves that you have mastery in the subject matter. The thesis demonstrates that you are capable of finding solutions to significant problems. It shows that you can perform critical analysis and make sound technical decisions based on the findings. Most importantly, the thesis is proof that you can describe the project related activities and results in a well written scholarly publication, which is your thesis.

1. Your thesis is published by the CSUN Library. It is available for inspection by anyone, throughout the world. Each graduate thesis bears the signature of this department. Therefore, your thesis must be written to a standard consistent with published technical work in professional publications, such as: conference proceedings, SIG publications, and scholarly journals.
2. A successful demonstration of the software product you have produced is clearly very important; a defense is not complete without such a demonstration. However, the demonstration is seen only by the committee. In reality, you earn your degree with the thesis, not with the demonstration. Many students spend more time and energy in getting the demonstration ready and not nearly enough in writing the thesis. That is a poor choice that often results in delayed graduation. It is important that you schedule enough time for writing the thesis.
3. By the time you start writing the thesis, you should have acquired sufficient writing skills in English. The preparation section, on the following page, provides some ideas on how you might accomplish this.

2. DISCLAIMER

This is a Work-in-Progress (WIP) product; it is not the final release. Whereas, the guidelines in this document provide useful information, it has not yet been approved by the computer science department. Until this document is approved and released by the department, please seek guidance and advice from your committee chair on how you should write your thesis.

3. PREPARATION

1. Writing your graduate project thesis is no simple task. It takes months of preparation and meticulous hard work. You need to work closely with your thesis advisor in getting the thesis ready for committee review and defense. It is not uncommon for a student to produce 4 to 5 drafts before arriving at a copy ready for distribution to the committee.
2. The thesis must be written in grammatically correct English and be easy to read. Do not expect your committee chair to copy edit your work. S/he is there to give you guidance on technical issues on thesis writing, such as: thesis outline and topics to be covered. S/he is not there to provide you lessons in writing English. If the draft you submit to the committee chair is not of reviewable quality, then s/he may return it for you to revise and resubmit.
3. If you are an ESL student, or you need to hone your writing skills, then consider taking writing classes. Passing the UDWPE alone does not prepare you for writing a thesis. Seek out classes on writing technical publications. Also look for online resources on thesis writing. The CSUN English department provides individual 30 minute writing review sessions on proper use of grammar and sentence structure. A graduate student can book one such review session per week. Check with the writing lab in the English department. Sometimes, you may find tutors who will copy edit your work for correct use of English. Do not ask the reviewer or any external consultant to write the thesis for you. That constitutes academic dishonesty. Any level of academic dishonesty can have severe consequences, including the need for you to start your project over with on a new topic and a new proposal.
4. Carefully consider the word processing and other utility tools, such as grammar and style checkers, that you will be using to write the thesis. Arrange to learn the techniques for using the tool effectively. Tool issues are addressed further in section 7.

4. FIRST, CREATE AN OUTLINE

1. Plan out the thesis chapters and create an outline listing the chapters you will have in the thesis. A suggested list of chapters appears below. This is preliminary; you can change the chapter list as the thesis develops. Suggestions for what should be covered in these chapters appear in Section 6, DETAILED STRUCTURE AND CONTENTS OF THE THESIS.
 - 1.1. Abstract: A summary of the objectives and accomplishments. Typically 1 page long.
 - 1.2. Objectives: Describe the problem that you set out to solve and the solutions you have achieved.
 - 1.3. Introduction: Describe the background of the project work. Establish the context. Discuss why this problem is important. Briefly describe the development process you will follow.
 - 1.4. Literature Review: Provide a survey and a critical review of related prior work.

- 1.5. Analysis and Requirements: Describe the problem analysis, enhanced with an analysis model in UML. Specify the resulting set of system level and software level requirements.
 - 1.6. Design: Describe the architectural design and the detailed design enhanced with UML model diagrams. Describe your rationale for the design decisions with supporting data collected from trade-off studies. Describe the specific tools and techniques used in subchapters.
 - 1.7. Implementation: Describe the implementation approach. Describe software reuse, design patterns, special coding techniques, etc. Describe special tools used, if any.
 - 1.8. Testing: Describe the testing approach. Describe sample test plans and test results.
 - 1.9. Tools and technologies: Describe the tools and technologies used in accomplishing the project in the context of the project activities. This can be integrated into topics 1.5 to 1.8
 - 1.10. Conclusions:
 - 1.11. Appendices:
2. Get the thesis outline approved by your committee chair.

5. MASTERY

1. Successful completion of a graduate Project demonstrates that you have the ability to analyze and develop solutions to a problem of significant complexity and stature. The work you undertake must be of much higher degree of complexity than the projects done in Comp380, 480, or 490 classes. Simply producing a software application, using a run-of-the mill method and an ad-hoc process, does not demonstrate mastery worthy of a graduate project. Through your project, you must demonstrate mastery of the current software engineering and computer science disciplines. Use of current techniques and technologies in completing the project work is important.
2. In writing the thesis, you need to describe the problems and the solutions in an organized and clear manner. You should use standard software engineering and computer science nomenclature. If in doubt, consult the IEEE Standard Glossary of Software Engineering Terminology (No 610.12-1990) or the Software Engineering Book of Knowledge (SWEBOK). Both these references are available online through the CSUN library.
3. Today, UML is accepted as the modeling language of choice in both computer science and software engineering. Therefore, use of standard UML is strongly recommended. Avoid using your own ad-hoc drawing conventions. If you don't know UML, then learn it. Attend a class, or take one of the many online tutorials available in the Internet. At CSUN, UML modeling is introduced in Comp380; advanced UML is covered in Comp586. Produce model elements that adhere to the core UML syntactical and semantic rules. This will allow the reader to read and interpret the meanings of the drawings in the analysis and design models.

4. Strongly recommend using a modern, UML capable CASE to create the UML model diagrams. A CASE tool will make it easy to create the model diagrams, maintain a consistent model, and will (often) guard against breaking UML syntactical and semantic rules. There are many free CASE tools available, such as: Visio-2010 via MSDNAA, ArgoUML, Visual Paradigm, and Poseidon. Find one CASE tool that you like, learn it well, and use it capture the analysis and design models. Use it as a CASE tool, not as a drawing tool, i.e., follow the rules of UML. Many CASE tools provide a capability for generating reports describing the model information captured in the tool. With some editing, you can incorporate these reports into your thesis.

6. DETAILED STRUCTURE AND CONTENTS OF THE THESIS

In general, a graduate thesis should have the following chapters and sections. Some chapters are mandatory; others will depend upon the nature of the work. Fill in the chapters in the thesis outline that you have already developed, as described in section 4. Each chapter should elaborate on one major concept, such as prior work, design, implementation, testing, tools used, etc.

6.1. Abstract

Although it appears first, plan on writing the abstract last. The abstract is the hardest part of the thesis to write, and it is the part most readers of the thesis will read it first. The abstract should be very well written. It should be clear, easy to read, and to-the point. The abstract conveys the most important messages regarding your project, such as: what you set out to do? How did you do it? What results were obtained? You will have a much better shot at writing a good abstract after you have completed all the other parts of the thesis.

6.2. Table of Contents (TOC)

1. The following TOCs are mandatory.
 - 1.1. General TOC: listing chapters, sections, and subsections to the lowest levels.
 - 1.2. List of figures.
 - 1.3. List of tables.
2. The page numbers in each TOC should be hyperlinked to their targets (sections, figures, tables). Hyperlinked page numbers should work even in a PDF format document. If you are using Microsoft Word/Open Office Writer then strongly recommend using the TOC generation tool.

6.3. Chapter on Introduction

1. This chapter is mandatory and, at a minimum, should cover the following topics:
 - 1.1. Introduce the reader to the particular problem your project is attempting to solve. Most projects have multiple objectives. For ease of cross referencing, it is a good idea to state these in a numbered list. The rest of the thesis describes how you have accomplished what you have described as the objectives.

- 1.2. In general, the objectives stated in the thesis should match those stated in the project proposal. If there are substantial differences then file a revised proposal with the GRIP. Although the GRIP procedures do not require it, you should advise all committee members of the changed proposal.
 - 1.3. Set the scene by providing a background for the work. Why is this work important or interesting?
 - 1.4. Write a summary of the overall approach. Include brief descriptions of the development process, design, implementation, and testing approaches.
 - 1.5. The tools and technologies you used should be mentioned here but described and discussed in later, in a chapter dealing with the technical details.
 - 1.6. Provide a synopsis of what the other chapters contain. These descriptions should be very brief, one or two sentences for each chapter.
2. By the time the reader has finished reading the Introduction s/he should have a clear understanding of the problem you set out to address and how it has been solved.

6.4. Chapter on Review of Literature

1. This chapter is mandatory and is different from the background provided in the Introduction. The background provides general information. The literature review focuses on issues that are more specifically related to the work in your project.
2. Describe similar work done by others in the past and described in the literature. If you cannot find prior work in the literature, then it is most likely that the work you are describing is too simple to qualify as a graduate project.
3. Your thesis needs to demonstrate that you have done a literature search and completed a critical analysis of the relevant literature describing prior work in the field. Demonstrate this by writing some discussions on what others have done, what they have achieved, and limitations of their work. If they exist, then provide reviews of prior work in the literature, this shows that you have done a comprehensive literature search.
4. Do not copy and paste text from the literature; paraphrase the contents in your own words. References must be cited here in the introduction and everywhere else in the thesis. Do not just provide a number like [23]. Say something about the work.

Example: Jones and Bartlet reports that use of Agile processes reduced mean development time in graduate projects by approximately 12% [23]. Kissinger opined that these results need verification with a wider sample. He pointed out that the Jones and Bartlet study was based on results from only six projects [32]. A follow-on study by Jones and Bartlet included over 30 graduate projects and substantiated the results reported in the original study [24]. An independent study by Reifer, involving 27 industrial projects, claims development cost saving of 13% attributable to the use of Agile methods [42].

6.5. Technical Chapters

In general, the chapters described in the following subsections are expected in a thesis describing a graduate project. Some of the listed chapters may not be applicable to your thesis, and additional chapters covering special topics may be needed. Seek guidance from your committee chair on the chapters your thesis should contain. A good approach is to describe each major concept/task in a separate chapter, and describe minor related concepts in sections/subsections within the chapters.

6.5.1. Chapter on Development Process

Describe the development process you followed. To demonstrate your mastery in software engineering and computer science, your project should follow a standard software development process, rather than an undefined or ad hoc process. Generally, a process framed on the agile development philosophy works well for graduate projects in software engineering and computer science. If you choose an agile process, then be sure to describe the process you followed for making the multiple deliveries and demonstrations to your committee chair or research group. Doing an iterative development and making multiple deliveries is a key practice in any agile process; if you did not do this then the process cannot be called agile.

6.5.2. Chapter on Analysis and Requirements

Describe how you did requirements elicitation, conducted the analysis, and arrived at the specified requirements. Provide analysis models, not just words. Some suggested model elements are: use cases, activity diagrams, sequence diagrams, and domain models. The analysis models should express the system architecture and the top level behavioral requirements. Don't provide a superficial model with just one or two context level use case diagrams.

6.5.3. Chapter on Design

Generally describe the architectural design model and the detailed design model in separate chapters. Always, discuss the alternatives considered and the rationale for the choosing the solutions you adopted. Describe the architectural and detailed design models in a disciplined manner using both text and comprehensive design models, ideally expressed in UML. Use of UML is highly recommended over using ad hoc or older modeling notations. Suggested UML design model elements are: class diagrams, interaction diagrams, structured classes, components, subsystems, and deployment models. Produce the model diagrams with a modern CASE tool, not drawing tools. Provide a comprehensive design model with sufficient design information, not just one or two top-level model diagrams. Note that to describe a design adequately you must describe both its static view and the dynamic view. The static view includes elements such as: classes with inheritance and aggregation, structured classes, interfaces, components, subsystems, and deployment. The dynamic view includes: activity diagrams, sequence or communication diagrams, and the state model, when appropriate. Remember that, in most projects, the design model is the main aspect of your work, and it deserves a good deal of your attention.

6.5.4. Chapter on Implementation

Describe the overall strategy for implementation tasks, such as incremental builds, risk mitigation measures. Discuss the reasons why you chose the specific programming language, development tools, testing tools, and the implementation platform. Discuss strategies for reuse of existing products and components. Use of design patterns in the implementation demonstrates sophistication in the subject matter and is highly encouraged. Generally, you do not need to provide source code in the thesis, unless that code is central to your thesis, e.g. if you created new design patterns and need to describe the logic of those design patterns using code. However, note that describing design logic using detailed design models demonstrates a higher level of expertise than using code to do the same.

6.5.5. Chapter on Testing and Validation

Describe how testing and validation tasks were performed. Describe the plans and strategies used in unit testing, integration testing and system testing. Address regression testing. Describe the test plans and provide test procedures for testing the critical functions.

Describe the test tools you used. Whenever possible, involve someone else, such as friends and colleagues, in the testing and verification process, and include their comments and observations.

Provide test metrics, such as number of defects found, defect density of the discovered defects, code and branch coverage metrics. Ideally there should be an analysis describing the defect injection and discovery characteristics, such as: types of defects, injection phases and discovery phases.

If your project serves an external customer then you must involve end users, selected by the customer, in the testing process. Examples of such projects are: community service projects, project from your place of work, or projects with an external sponsor. For graduate projects involving the end users in the testing serves as an acceptable validation process.

6.5.6. Chapter on Tools and Technologies Used

Describe any state of the art tools and technologies you used in the project. **THIS SHOULD NOT BE A TEXT BOOK LIKE DESCRIPTION OF THE TECHNOLOGIES.** Filling up a thesis with descriptions of tools and technologies that are readily available in books or published literature does not add any value to the thesis. You must provide some discussions that demonstrate that you have performed some critical analyses of the subject matter.

It is important that you describe how the tools and technologies are being applied to the project you have completed.

You should include some discussions on evaluation of alternate tools and techniques, provide comparisons and state rationale for choosing the ones you did.

6.6. Conclusions

In the conclusion chapter summarize the problem you set out to solve, describe what you have achieved, and prospect for future work.

1. Refer back to the problems you encountered and how you overcame those, or found workarounds. Always refer back to the main body of the thesis for the detailed descriptions; the conclusion section should not contain detailed descriptions of the problems or the solutions.
2. Address how you have met the original objectives of the project (i.e. proposal contents).
3. Discuss potential future work.

6.7. Appendices

Use Appendices to present material that will interrupt the flow if included in the main body of the thesis. Typical contents of appendices include: Code, data tables, detailed analysis and design models. If a user manual is called for, then provide it in an appendix.

6.8. Bibliography

1. Every citation made in the body of the thesis must appear in the Bibliography. Similarly, every item listed in the Bibliography must be cited in the body of the thesis.
2. The committee may use the list of references as a yard stick to assess how well you have researched the field before setting out to do your project. The committee may look for completeness and also accuracy of the references. Error in the bibliography will need to be corrected before a thesis is approved.
3. Follow a single standard method for citing and listing both the print references and the online references. There are many different formats for citing and listing references, such as: APA, MLA, ACM style, IEEE style, etc. Choose one and follow it consistently throughout the thesis. Note that there is a standard method for listing online references, listing just the URL is not sufficient.
4. Today, most print publications, such as journal articles and conference papers, also exist in the web, typically in digital libraries or in the author's web sites. When listing references to such printed sources, provide references to the original printed source, not the online sources.
5. In general, publications that are not peer reviewed, such as blogs, are not credible sources of reference. Citations from Wikipedia are marginally acceptable but should be avoided if possible. This is because the Wikipedia review process is not as well controlled as in professional journals and in proceedings from conferences organized by professional institutes. Technical publications from well established and recognized organizations like IBM, Microsoft, Apple, Oracle, Motorola, etc. are generally acceptable.

6. You should have read every referenced item you list in the bibliography and every item in the bibliography should be appropriately cited in the body of the thesis.
7. If you used the Citation and Bibliography Tools, then you can auto generate the bibliography. This will save you a lot of work and time.

6.9. Other Sections

The following sections are highly desirable, because they greatly improve the readability of the thesis.

1. Glossary of terms.
2. List of abbreviations.

7. USE A MODERN WORD PROCESSOR AND A STYLE BASED DOCUMENT TEMPLATE

1. Unless your committee chair directs you to do otherwise, strongly recommend using a modern word processor such as Microsoft Word 2007 or a later release. Microsoft Word provides comprehensive facilities for writing long documents that are hard to beat. The really useful features include: planning an outline for managing chapters, sections and subsection; automatic saves (say every 10 minutes) and recovery, versioning. Ability to use styles separating contents from presentation. Tools for automatic generation of the table of contents, the bibliography, and a glossary. Ability to cross reference items within the document that can be updated with a single click. Ability to insert foot notes and end notes that migrate automatically as the document changes.
2. Microsoft word also has powerful tools for electronic reviewing, change tracking, and merging of comments from multiple reviewers.
3. Another advantage of using Microsoft Word is that many CASE tools provide automated document generation. Typically, these documents are produced in Microsoft Word format, allowing you to edit the auto generated text and drawings ready for insertion into the thesis.
4. How to obtain Microsoft Office 2010
 - 4.1. Microsoft provides free cloud based access to Office 2010 through its Windows Live SkyDrive technology.
<http://explore.live.com/windows-live-skydrive> Retrieved June 8, 2011
 - 4.2. Student editions of Office 2010 can be bought and downloaded for home use from many Microsoft authorized vendors, or directly from Microsoft. The current cost is \$79.99. Search for Microsoft Student – Office 2010
<http://www.microsoft.com/student/en/us/software/office-2010.aspx> Retrieved June 8, 2011.
 - 4.3. Most department and university computers in the labs provide Microsoft Word; you can use these to write your thesis.

- 4.4. As an alternative, you can try using Open office Writer, which provides compatibility with Microsoft Word. However, some of the advanced features are not compatible. Search for OpenOffice.org Download <http://download.openoffice.org/> Retrieved June 8, 2011.
5. Strongly recommend using a document template. You can create a style based thesis template from scratch; however you can save time by reusing a template from an existing thesis. With your committee chair's permission, get a copy of an existing thesis that has used a style based template. Create a skeletal template for your thesis by deleting all existing contents, but keeping all the style elements. Start writing your own thesis using this new template. You can change the template as the thesis develops.
6. Strongly recommend using the Microsoft Word **style** feature. Use of **styles** is similar to using Cascading Style Sheets (CSS) in creating web pages. **Styles** separate the concerns of formatting from the contents. Use **styles** when creating section headings, figure captions, table captions, lists, tables, and of course the text in the thesis. If you write the thesis using a suitable template that uses **styles** then you will be able to built in tools in Microsoft Office to generate the table of contents, the list of figures, the list of tables, the glossary, and even the bibliography in APA or MLA format.
7. In choosing a tool, you should also determine the committee members' needs for reviewing the thesis. Members who do electronic reviews may insist on getting a thesis in Microsoft Word or Open Office Writer format.
8. Please note that it is very cumbersome to do electronic review of documents in PDF format.

8. GENERAL GUIDELINES FOR THE THESIS CONTENTS

1. There is no rule governing the length of a thesis. A thesis does not have to be 60 pages long! Do not "fill-up" a thesis with irrelevant material just to make it long.
2. Do not mix unrelated issues in the same chapter. For example, in the design chapter, do not discuss testing issues. One exception to this is when arguments are needed to justify actions related to the current topic.
 - 2.1. Example: When a specific design decision was made to simplify some implementation related problem. In such a case, it is reasonable to discuss the relevant implementation issues in the design section.
3. In a thesis, every sentence should have a purpose and every sentence should say something relevant to the project work. Be succinct in what you write. Avoid vague rambling text.
4. Your thesis is a piece of scientific work. It should not contain any unsubstantiated opinions or claims. Every technical claim made or result cited must be substantiated with credible references, or result of some reasoned discussion contained in the thesis.

5. A thesis is not a platform for making speculative comments, or for making predictions that are not grounded in results from published scientific work.
6. A thesis should not have any tutorial text; it is not a text book. If a user manual is called for, then provide it in an appendix.
7. Throughout the thesis use standard software engineering or computer science terminology. Provide definitions of uncommon terms in a glossary. See section 6.9
8. Use a consistent naming convention. Do not refer to the same concept or “thing” by different names in different parts of the thesis.

9. GENERAL FORMATTING GUIDELINES

1. Read and follow the style guidelines produced by the Graduate Studies, Research and International Programs (GRIP) office. The GRIP guidelines focus on the form, not the contents and have some strange rules, but you need to adhere to those rules.
2. Provide headings for all chapters, sections and subsections and prefix each heading with a section number. This is easy to do if you use a style based template in Microsoft Word or Open Office Writer.
3. For improved readability the headings should be left justified rather than center justified.
4. Number all figures and tables. It is best to use a localized numbering scheme where the figure/table numbers are prefixed with the section number. This makes it easy to locate the item when referenced from elsewhere within the document.
 - 4.1. Examples: The two first figures in section 2.0 will be numbered Figure 2.0-1 and Figure 2.2-2. Similarly Figure 3.1-3 would be the 3rd figure in section 3.1. Table 3.3-3 would be the 3rd table in section 3.3.
5. Provide captions for all figures and tables. Example: Figure 2.3.1: The System Context Diagram. It is common for figure captions to appear below and table captions to appear above.
6. If you are using Word or Open Office Writer, then use **styles** to mark the figure and table captions. This will enable you to auto-generate the List of Figures and the List of Tables.
7. Figures, tables, and their associated captions should be centrally justified.
8. If you are providing a list of items, then use a numbered list, not bulleted list. It is easier to reference an item in a numbered list.
9. Follow a consistent method for citing references such as: the APA, MLA, ACM style, or IEEE style.
 - 9.1. If you are using Microsoft Word, or Open Office Writer, then consider using the **Citation and Bibliography Tool**. You will then be able to generate the bibliography automatically in APA or MLA style.

- 9.2. Strongly recommend using the Word/Write Cross References Tool for managing all internal references. You will then be able to automatically update the cross references even as the thesis develops and the cross referenced items change their numbers.
- 9.3. Use of these tools saves a great deal of time and results in a thesis where the cross references and reference citations are more accurate.
- 9.4. There are many free online tutorials on how to use these tools; find one and learn to use them.
10. When cross referencing, instead of writing “As previously stated ...”, provide a cross reference number to the referenced section, figure, or table; one more reason to use the (Word/Writer) Cross Referencing Tool.
11. Expand all acronyms on first use. It is a good idea to list all acronyms in a list of abbreviations. See section 6.9.
 - 11.1. Example: Graduate students should be familiar with the Electronic Thesis Document (ETD) system.
12. All figures and tables should be referenced in the body of the thesis using their captions.
 - 12.1. Be mindful of the language you use. When referencing a figure, write, “The initialization logic is illustrated in Figure 3.2-1: System Initialization” rather than “Figure 3.2-1 illustrates the initialization logic”. Why? A figure, by itself, does not “illustrate anything”, rather “something is illustrated in a figure”. Use a similar language when referencing a table. Picky, but precise.
13. If you need to include diagram or illustration copied from some other document, then cite the source with a reference number in the Figure caption and also in the text that introduces the diagram or illustration.
 - 13.1. Example: Harper reports the results from a study that examined the relationship between software structural complexity and software quality [31]. The relationship is illustrated in Figure 3.4-1: Design Complexity and Residual Defect Density.
 - 13.2. The figure caption would read

Figure 3.4-1: Design Complexity and Residual Defect Density [31].
14. Do not copy and paste diagrams that are blurry. Redraw the diagram instead. You should cite references, even if you redraw the diagram, unless you make substantial changes. This guideline applies to all cut-and-paste items, including graphs, and tables.
 - 14.1. Do not include any art work, such as photographs, that are blurry, or difficult to interpret.

10. ELIMINATE PLAGIARISM

1. There must be no plagiarism in a graduate thesis. The computer science department has zero tolerance against submission of any work containing any amount of plagiarized material. ***Follow the guidelines stated in the CSUN catalog in the section named "Academic dishonesty"***. These guidelines preempt any other ideas that you might have on what constitutes plagiarism. Please follow the rules stated there, especially on quoting sources and paraphrasing contents written by others. Consequences of submitting plagiarized work is very serious, including starting the project over with a different topic.
2. Do a similarity check of your thesis using www.Turnitin.com and use it to eliminate all plagiarized material. Ask your committee chair for a Turnitin account. With quoted text and bibliography excluded, the similarity index should be less than 10%. Turnitin sometimes flags URLs as plagiarized text, but you can ignore those findings.

11. WRITING STYLE

1. Write short sentences in active voice in plain language. Each sentence should address a single topic.
2. Write short, focused, and coherent paragraphs. Each paragraph should have a clearly stated topic sentence that describes the main issue covered in the paragraph. The rest of the sentences should provide discussions on the topic sentence, or answer the question raised in the topic sentence.
3. Read each paragraph **aloud** and listen. Does it sound right? Is the sentence structure clumsy? Is it difficult to understand? Listen to the sentences as you read them aloud- would you speak like this when making a formal presentation in class? Revise and edit as needed.
4. Do a grammar and style check using one of the tools listed in section 13

12. GENERAL ISSUES

12.1. Critical Analysis

Critical analysis skills are taught in all graduate level classes. The thesis demonstrating your culminating experience must exhibit your ability to do critical analysis. Evidence of critical analyses can be placed in a separate chapter. Alternatively, the critical analyses can be integrated directly into the various technical chapters describing analysis, design, implementation, and testing. In the technical chapters describe how you considered alternatives and selected the ones you did. Give rationale for the choices you made. A master's thesis must demonstrate that you are capable of making sound technical decisions based on results of critical analysis. Describe any technical problems you encountered and how you overcame those or found workarounds.

12.2. Take Credit For Your Work

Throughout the thesis, and especially in the chapters describing analysis and design, clearly state what you have achieved. Take credit for successes; describe lessons learned from unsuccessful outcomes, including change of plan/approach etc. Address what would have happened if you were to use a different tool or technique. Address what problems were caused by the need to use legacy products, if any. When appropriate, describe problems caused by the need to use specific tools, platforms, hardware, etc. With the benefit of the hindsight, discuss how would you do things differently. Suggest future work.

13. SOME USEFUL RESOURCES

(This part is incomplete; I would welcome suggestions on additional resources)

You may find some of the following assets helpful. The commercial products are provided for information only; none are specifically endorsed or promoted.

Style and Grammar Checker

1. "Writing Enhancement Software review" Retrieved: June 8, 2011. A good source for checking English grammar and writing style
<http://writing-enhancement-software-review.toptenreviews.com/>
2. WhiteSmoke Grammar tool
<http://www.whitesmoke.com> Retrieved June 8, 2011
3. English Software
<http://www.englishsoftware.org/> Retrieved June 8, 2011

Books and Articles on Thesis Writing

1. Trimble, John R. Writing With Style: Conversations on the Art of Writing (3rd. Edition). Longman, New York: 2010
2. "Purdue Online Writing Lab" Retrieved: June 8, 2011
<http://owl.english.purdue.edu/owl/>
3. Zinsser, William. On Writing Well, 30th Anniversary Edition: The classic Guide to Writing Nonfiction. Harper Collins, London: 2006.
4. "A Research Guide for Students" Retrieved: June 8, 2011.
<http://www.aresearchguide.com/styleguides.html>
5. University of Waterloo guide on Thesis and Long Documents. Retrieved June 8, 2011
<http://ist.uwaterloo.ca/ew/thesis/ThesisCourse/WordTheses.html>

Learning Tools

1. A thesis template with styles document that can serve as a starting point; it must be modified to match CSUN requirements. Retrieved: June 8, 2011
<http://csce.uark.edu/~cwt/ADVISING/MS-thesis-template.doc>
2. Tutorial on creating numbered section headings etc. in Microsoft Word. Retrieved June 8, 2011
<http://www.shanakelly.com/word/numbering/numbering20072010.html>

14. CSUN ELECTRONIC THESIS DOCUMENT

There is a system for online submission of proposals and the final thesis document to GRIP. You will need to register to take advantage of these facilities. Details in <http://etd.csun.edu>

15. COMMENTS AND SUGGESTIONS ON THIS GUIDELINE

Comments and suggestions on this guideline are welcome. Please email to shan@csun.edu