

General Documentation Style Guidelines and Standards

The following list provides the basic document organization, required information and format. Each item under a bullet "•" should be considered to be required as a major element or section in your final reports (and/or artifacts that are deliverables)¹. The final governing document is however the tailoring guidelines specific to the artifact in question.

In general use the following rules of style as a guide. Use Times Roman or Times New Roman (use *Courier* when proportional spacing is necessary or preferred (e.g., listing of algorithms and or code segments) to indicate nesting.). Use 10 - 12 point font 1.5 spaced. Use indents on all paragraphs except the paragraphs immediately following a section heading (e.g. **1.2.3. Requirements**). Try not to leave large blocks of white space. Number the figures consecutively like "Figure 1. This is a caption." (the next figure would be 2 and so on). Place figure captions *under* the figure and tables in exactly the same way except place table captions on the *top* of the table. All section headings should not be indented (i.e., they should be flush to the left margin). Margins should be one inch (top, bottom, left and right). Use page numbers on the bottom (footer) and centered. Do not use a page number on the first page (it is preferable that the front matter (title page, abstract and frontmatter) be page numbered in roman numerals (e.g., i, ii, iii, iv, ...) but its not a strict requirement. Used justified text as is done in this paragraph.

- **Title page** which includes course name and number, assignment/deliverable name, date, and team name (and/or team member names). The title should be at least 28 point font and center every thing.
- **Abstract page** which is necessarily brief, includes a description of problem, procedures used for your solution and a brief discussion of your results. You may wish to put important points in bullets. Use the word Abstract at the top of the page (bold and centered).
- **Front matter** which includes the following (be sure to use leaders and *center* the title):

Table of Contents

List of Figures and page numbers

List of Tables and page numbers

- **Introduction** which includes the following:

Goal or problem statement: For example, these guidelines identify how to prepare a requirements specification, design and testing package as a deliverable (i.e., auditable, configurable, traceable and maintainable) document. This part will be derived within the framework of the target application (i.e., based on the purpose of the particular artifact [e.g., SRS, Design Notebook, Test Report, User Manual]).

Scope: covers methods, assumptions (or constraints) and procedures. If a team decides to take exception to any of the stated requirements (or standards) then it is mandatory to obtain authorization from the instructor (signed waiver to be included in your artifact) who should be viewed as your customer.

Objectives and milestones: Objectives are the tasks which must be completed to obtain a goal. Milestones are objectives except that they are correlated with time. A preliminary review is advisable for each artifact (i.e., show your template or the actual artifact to the customer [instructor] before it is actually due to make sure it meets his expectations). Include a list of the artifacts and a schedule showing their time span (begin/end dates) and due date. It is suggested that all artifacts be inserted into one schedule that can be reused with all deliverables..

¹ All deliverables should be walked through by team members to find errors and make suggestions for improvement.

Organization of the document: This section *must be included* in each artifact. The introduction gives high level and administrative information. The second section is an optional background section that describes terminology and notation. The third section is your design representation. The fourth section is results (i.e., scenario traces and engineering tradeoff tables). Conclusions follow the results, where you subjectively discuss what you observed, and learned. Finish with the references and appendix.

Other: [optional] information you want the reader to know before starting such as (1) exceptions to this format and justification (see "Results" below), (2) glossary and location, (3) acronyms and location, (4) references and location, and (5) Appendices (e.g., figures are all found in Appendix A). Label this subparagraph as is appropriate to its content.

- **Background** which is optional and includes terminology (e.g., buzzwords or terms not covered in the text) and non-standard notation (i.e., not covered in the text). Here is the place to describe any exceptions (or waivers) to the standard.
- **System Representation and Results** which includes (either or both) the following items:²

Part I *Reverse Engineering* [optional]

1. Functional English Description
2. Formal Statement of Requirements
3. Design (e.g., DFDs, P-specs, Structure charts, DDs)
4. Source code of critical algorithms may be listed by module (ensure adequate comments [including preambles] for easily relating legacy code to the design and subsequent new code). However it is not advisable to turn in a huge document that contains mostly source code listings.

Part II *Forward Engineering*

1. Software Requirements Specification
2. Design Notebook
3. Abbreviated Test Report³ and User Manual.
4. PDR/CDR and Demonstration and Source code (deliverable in electronic form).

Results can include the following items and will vary according to the target application and/or artifact (i.e., Part I versus Part II). Results should be handled as a separate subsection. What was achieved with respect to the goals of this artifact or phase in the life cycle. What was not achieved and why. How might the results be improved.

- **Requirements Traceability** which includes a mapping of the requirements in this document to the requirements applicable within the design [notebook] specification and the implementation (i.e., code). The traceability matrix should show a mapping from the formal requirements, to the design (i.e., use the DFD numbers and names here) and finally to the code (use module names [all that apply]) to designate what actually implements a given requirement. Finally, make sure you define how the requirement should be verified (i.e., 1=demonstration, 2=inspection, 3=analysis, or 4=analogy).
- **Conclusions** which are your subjective observations (e.g., tables, graphs, file summaries, output file summaries). This is a very important contribution to the document. You should have some discussion of why

² Each item will be a subsection of the "System Representation" section (with the exception of source code which fits more appropriately in the appendix). If your document requires the delivery of source code then it should be included in an Appendix (unless a waiver from the instructor is obtained for say just providing an electronic version).

³ Must describe the plan, issues and results of integration.

certain decisions were made (e.g., partitioning of modules, what data structures were used and why, and reporting format [program output]).

- **References** should be consistent from one to the next. Use the following example as a guide.
 - [1] Gopinath, P., and Gupta, R., "Applying Compiler Techniques to Scheduling in Real-Time Systems", *IEEE Proc. of Real-Time Systems Symposium*, pp. 247-256, 1990.
 - [2] Haban, D., and Shin, K.G., "Application of Real-Time Monitoring to Scheduling Tasks with Random Execution Times", *IEEE Trans on SE*, pp. 1374-1389, December 1990.
 - [3] Jensen, E.D., and Locke, C.D., and Tokuda, H., "A Time Driven Scheduling Model for Real-Time Operating Systems", *IEEE Real-Time Systems Symposium*, pp. 112-122, 1985.
 - [4] Chung, J.Y., Liu, J.W.S., and Lin, K.J., "Scheduling Periodic Jobs that Allow Imprecise Results", *IEEE Trans. on Computers*, Vol.19, No.9, pp. 1156-1173, Sept. 1990.
- **Appendix (ies)** Are lettered A -Z and may contain (1) expanded examples or notes, and (2) diagrams may go here (e.g., key view graphs showing changes made from PDR to CDR), and (3) source code. You may not use a number in front of the word APPENDIX and do not forget to title each appendix as shown below (note that the whole title is centered).

APPENDIX A: Documentation Standards

1. Scenarios (external system transactions) are used in test descriptions. They are used to describe I/O in relation to the operational profile (see the glossary).
2. Graphic nodes (DFDs and the like) should be hierarchically numbered. In the text each figure must be discussed and referenced (i.e., explain your diagrams). Use the specific figure number as the references (e.g., "... as shown in Figure 5.")
3. Section/Paragraph numbering. This is especially tricky. Use an outline to make sure the numbers make sense and that each topic is properly separated (and subordinated). In this way, the whole document will make more sense. Thus paragraph numbering will be limited to three levels of indenture (e.g., ¶1., to ¶1.1, 1.2, ... , ¶1.2.1, 1.2.2, 1.2.3, ...). After that you'll have to use bullets, bold heading and the like. The reason is because the table of contents (TOC) becomes too unwieldy. Use your word processor's ability to automatically generate the TOC. This means that you must use a word processor that can provide this capability.