DOCUMENTATION PLANNING GUIDE

Revision 2 - by Russ Tront 95/05/10

The purpose of this document is to give CMPT 275 students some guidelines for planning release documentation. This is necessary as students don't appreciate that without some planning on a multi-person, multi-document project, the result will be poor organization and lots of inconsistencies. This will result in loss of marks on the assignments. Thus your team MUST adopt a formatting standard that is very similar to this one, and you must all stick to it.

THE RELEASE PAGES

All computer software configurations items (CSCIs) must be coordinated. The user manual must match the requirements specification and the current source code module revisions. Otherwise, the software might not do exactly as the manual shipped with it says it will do. Similarly, it is possible for an updated main program to call an old library module that no longer supplies up-to-date enough services, thus resulting in unpredictable execution. As a result, the 'set' of documentation and code items which make up a software system must be assembled together in a coordinated manner so as to mate properly at link and delivery time. They are issued in a package called a 'Release'. See Tutorial #2 for further info and some explanatory diagrams. In CMPT 275, each assignment is a release (though you may have an test release of your system between Assignment 4 and 5 which is not handed in). Assignment #1 is composed of only one item, the Requirements Specification. But every release is prefaced by three extra pages, and followed by one or more complete CSCIs. The three pages are:

- the release title page
- the release history page
- the release table page

In the first assignment, these are followed in a binder only by the Requirements Specification (including its own title and revisions history pages). Later, multiple CSCIs will be in the binder, each separated by a labelled tab divider. This applies to source code too; each module must be separated (though you don't have to separate the individual halves of a definition/implementation module pair). Additionally, the code listings should be in the same order as in the release table: main first, followed by the rest in alphabetical order.

Don't forget these first 3 items, and don't get them mixed up with the first few pages of the requirements specification document. In particular, make sure you understand the difference between a release and a revision!

The **Release Title Page** looks like the following:

Cmpt 275 Group F

Highway Software Product's

Gee Wiz System

Release 5B

Approved by:

Russ Tront, Project Manager _____

Marketing Dept. _____

Quality Assurance Dept. _____

The **Release History Page** looks something like the half-page below. You may use chronological or reverse chronological order, but pick one order and stick to it throughout the project. Each release should have a historical comment (or whole paragraph if necessary)

briefly describing what changed and why.

RELEASE HISTORY PAGE

Rel. 1	- 93/01/02 - was composed of just the preliminary Requirements Spec.
Rel. 2	- 93/02/03 - Updated the requirements spec, and added the User Manual and prototype code.
Rel. 3	- 93/03/04 - the architectural design was completed and libA (prototype) code thrown away.
Rel. 4	- 93/04/05 - the source code was completed and linked for initial in-house testing
Rel. 5	- 93/05/06 - the bugs found during in-house testing were found, and the user manual and 1 of the modules were updated.
Rel. 5X	- A variant of the system was made for Customer X. Rel. 5 continues to be current and shipped to most customers.

The **Release Table Page** would look something like the following. Each column specifies the revision number of each element in the set of items needed for a particular release.

RELEASE TABLE PAGE

Configuration Item:	Revision Number of Each Configuration Item in:					
	Rel. 1	Rel. 2	Rel. 3	Rel. 4	Rel. 5	Rel. 5A
Documents:						
Requirements Spec	1	2	2	2	2	2X
User Manual		1	1	1	2	2
Architectural Design			1	1	1	1X
Modules (Main first, then alphabetical):						
GeeWiz.mod		1	4	4	4	4X
libA.def		3				
libA.mod		4				
libB.def		1	2	3	3	3
libB.mod		1	3	4	4	4
libC.def			1	1	2	2
libC.mod			2	2	3	3
Other Files:						
makefile				1	1	1X
demoData					1	1

In effect, the Release Table acts as the super table of contents showing every CSCI in the release.
Note: In the C or C++ language, header files which include the function prototypes would replace the definition module files.

INDIVIDUAL DOCUMENTS

Each individual document or source code file listed on the release table is a CSCI. As a result, each has its own revision number and revision history. In code, this is in the form of revision history comments near the top of the file. Note that individual procedures don't have revision numbers as it is files that are the pieces that are compiled and linked together to form a release executable. Documents, on the other hand, typically have a title page, and revision (not release) history page as described below.

One of the big problems that have been noted in the past in CMPT 275 documents is student's lack of appreciation for consistency in fonts, sizes, bold, and underlines. Simply agreeing on the look of the text is all that is needed. If not, we find the following kinds of inconsistencies in the section headings from one document done by 4 different students:

1.1 Making Pancakes
2.1 - MAKING COOKIES
3.1 Making Candies
SECTION 4.1 Making Cakes

The first heading is 10 point size, a serif font with proportional spacing, underlined, no bold, first letter capitals. The second heading is all capitals, and has a different word spacing (or tab) and a "-" character between the section number and title. The third heading is only partly underlined, and uses a font without serifs (i.e. san serif). Finally, the last heading has the word SECTION in all capitals, bold but not underlined, is a larger font (16 point), uses a fixed-space font where the 'i's are the same widths as the 'm's (c.f. a proportional-spaced font), and the section name uses only first letter capitals. This non-uniformity in a document is totally unacceptable in both a commercial document and a CMPT 275 document. It can be avoided simply by a 5 minute planning meeting. I don't care if the team doesn't have exactly the same fonts on each of their wordprocessors/printers, as long as you choose similar characteristics:

-font size
-font serifs or not
-font character spacing technology (i.e. fixed pitch or proportional spaced)
-bold
-underline.

Also be careful of:

-word spacing
-tabs
-capitalization convention
-completeness of underlining
-header and/or footer format (if present)
-page, figure, and table numbering (if present).

A document doesn't have to have all the headings uniform, but at least all the ones at the same hierarchical level should be the same. In other words, section headings can be large, bold, capitalized, underlined, and start at the top of a new page. Sub-section headings could be medium size, bold, only first letter capitalized, and underlined. And sub-sub-section headings could similar but not bold.

Each individual document will have a **document title page** with title, revision, authors, and room for approval signatures. An example is:

Cmpt 275 Group F

Highway Software Product's
Gee Wiz System

Requirements Specification
Revision 2

Written by:

Russ Tront _____
Bill Smith _____

Approved by:

Quality Assurance Dept. _____
Marketing Dept. _____

The **Revision History Page** (not to be confused with a Release History Page!) should be the second page of an individual document. Revisions could start at 0 or 1, and could be listed in chronological or reverse chronological order. Be consistent among all the written documents.

DOCUMENT REVISION HISTORY:

Revision 0

-original 93/05/18 by Russ Tront and Bill Smith

Revision 1

-modified 93/07/19 by Russ Tront
-revised delete operation requirements in Section 2.1
to handle the new customer who needs to keep his
disk space used low.

Revision 2

-modified 93/09/21 by Bill Smith
-changed Figure 2.3 to reflect the changes needed as recorded
in the minutes of the 93/08/01 design review meeting.

The **Table of Contents Page** for an *individual* document in a release should look something like the following:

TABLE OF CONTENTS:

Generally, the format of the section lines should be similar to that used in the actual document (though you might want to remove underlining). Page numbers are optional in CMPT 275, as their absence makes it easier to merge outputs from several team member's independent work.

Some manuals have a special section which introduces the *reader* to the conventions that are used throughout the text for denoting certain things *of interest to the reader*. For example, computer outputs and prompts shown in a manual are generally in a fixed space font such as `courier` (since old printers and current dumb terminals only have fixed spacing, and we now associate that characteristic with computer outputs). Generally, what the user types in response to prompts is in a slightly different font, say a bold fixed space font like **`courier bold`**. And, there is often a way of denoting a keyboard key, rather than the typed characters themselves; for instance <cntl-c>, <esc>, <enter>, <cr>. These conventions are often briefly introduced in a separate section right after the table of contents. The 'Conventions for the Reader' section does *not* specify conventions for the authors such as section numbering formats and stuff like that. A reader convention section is only to help the *reader* understand that subset of the author's conventions that are relevant to the reader. Nonetheless, the author's should agree beforehand what reader conventions they will use while authoring documents.