***Coding Project Final Report***

**

***A Sample Document for   
Generating Consistent Professional Reports***

***Prepared by***

***John T. Bell***

***for use in* CS 440**

**at the**

**University of Illinois Chicago**

**September 2018**

### *****REMOVE OR REPLACE ALL TEXT IN RED ITALICS BEFORE SUBMITTING REPORT*****

### ***How to Use This Document***

*This document is intended as a sample template that can be copied and edited to suit a particular software engineering project. It was assembled from a combination of documents [1], [2], and [3].*

***Styles***

*This document was written in Microsoft Word, and makes heavy use of styles. The styles dialog is initially located on the menu bar under the “Home” tab in MS Word. It is recommended that the styles dialog be pulled off into a separate window when working on formatting of the document. If each paragraph is assigned a style, then modifying that particular style will affect all paragraphs in the document having the same style.*

*The table of contents uses the document headings and sub-headings to automatically generate table of contents information.*

***Tracking Changes and Multiple Authors***

*The “Review” tab in MS Word contains several tools that are of particular use when editing large documents, particularly when multiple authors are involved:*

*The “Tracking” section allows you to track the ( proposed ) changes to a document, and to step through each proposed change to either accept or reject the proposed changes.*

*The “Compare” section allows you to merge changes proposed by different authors, ( which will be marked in separate colors for identification ), and then to use the change tracking tools described above to accept or deny each change.*

*The recommended procedure is to start with each author having a copy of a base document, ( possibly this template. ) Then each author changes the section(s) they are responsible for, and submits their changed version to one person who acts as the overall document editor. This author merges the changes, selectively accepts or rejects each change, and then distributes a new base document to all authors for the next round of changes. It is also possible to merge the changes and then distribute the document, so that all authors can review the proposed changes. ( The latter approach may be appropriate for documents such as bylaws, in which the changes must be approved by a committee or a vote before they can be accepted. )*

***Dealing With Material that is Unwanted ( Right Now )***

*Much of this document includes material that is not needed for every project, and/or which may not have been written yet, and so should be removed before printing or distributing the document. There are several ways to do this, however each has their drawbacks:*

1. *Delete the material completely. The drawback is that now it is completely gone, and the only way to get it back is to copy it from some other document, if that is even available.*
2. *Change the font to "Hidden". The drawback is that this does not affect the numbering of sections, either in the text or in the table of contents. However the original style information is retained, so when the text is unhidden, individual paragraphs do not need to be restyled.*
3. *Change the style to "Hidden". This does cause the document to renumber properly, ( because the paragraphs are no longer numbered paragraphs ), but all the original style information is gone, so if it is unhidden later, then all the styles of all the unhidden paragraphs will have to be restored manually.*
4. *Move the text, say to the end of the document, and then do not print or distribute the material at the end of the document. If this is all that is done, then this material will continue to appear in the table of contents. However if the font is also changed to "hidden", then the material will not appear in the TOC, but all the original style information will be retained. The drawback to moving the text is that now you need to keep track of where the information was moved from, in case you ever want to move it back.*

*Hidden text can be made visible by selecting the backwards "P" paragraph mark on the "Home" tab.*

***Table of Contents***

*The table of contents lists the sections of the document and the page upon which each section starts. The table of contents may or may not include subsections, etc.*

*Microsoft Word ( and many similar programs ) has the ability to generate a table of contents automatically from section headings, and to update it when the document changes. The table of contents included in this template is automatically generated by MS Word, and can be updated by selecting the table and clicking on the menu that appears above the table at that time.*

***Note: Remove all instructional materials before finalizing and submitting this document, including this entire page and the one before it.***

**Table of Contents**

[*REMOVE OR REPLACE ALL TEXT IN RED ITALICS BEFORE SUBMITTING REPORT* 2](#_Toc67902775)

[*How to Use This Document* 2](#_Toc67902776)

[List of Figures 6](#_Toc67902777)

[List of Tables 7](#_Toc67902778)

[I Project Description 8](#_Toc67902779)

[1 Project Overview 8](#_Toc67902780)

[2 Project Domain 8](#_Toc67902781)

[3 Relationship to Other Documents 8](#_Toc67902782)

[4 Naming Conventions and Definitions 8](#_Toc67902783)

[4a Definitions of Key Terms 8](#_Toc67902784)

[4b UML and Other Notation Used in This Document 10](#_Toc67902785)

[4c Data Dictionary for Any Included Models 10](#_Toc67902786)

[II Project Deliverables 11](#_Toc67902787)

[5 First Release 11](#_Toc67902788)

[6 Second Release 11](#_Toc67902789)

[7 Comparison with Original Project Design Document 11](#_Toc67902790)

[III Testing 12](#_Toc67902791)

[8 Items to be Tested 12](#_Toc67902792)

[9 Test Specifications 12](#_Toc67902793)

[10 Test Results 13](#_Toc67902794)

[11 Regression Testing 13](#_Toc67902795)

[IV Inspection 13](#_Toc67902796)

[12 Items to be Inspected 14](#_Toc67902797)

[13 Inspection Procedures 14](#_Toc67902798)

[14 Inspection Results 14](#_Toc67902799)

[V Reccomendations and Conclusions 14](#_Toc67902800)

[VI Project Issues 14](#_Toc67902801)

[15 Open Issues 14](#_Toc67902802)

[16 Waiting Room 15](#_Toc67902803)

[17 Ideas for Solutions 16](#_Toc67902804)

[18 Project Retrospective 16](#_Toc67902805)

[VII Glossary 17](#_Toc67902806)

[VIII References / Bibliography 17](#_Toc67902807)

[IX Index 18](#_Toc67902808)

### ****List of Figures****

*( The title above is formatted as Heading 3, so that it appears in the table of contents, but was then modified to be centered and include a page break before the paragraph. Likewise for the List of Tables heading on the next page.. )* ***Note: Remove this instructional paragraph.***

*If a document contains a large number of figures, then it is appropriate to include a list of figures at the beginning of the document, following the table of contents. Each figure should include a title, and be numbered in a consistent logical fashion. The following list of figures was automatically generated from figure captions ( see* ***Error! Reference source not found.*** *on page* ***Error! Bookmark not defined.*** *), and can be automatically updated by right-clicking on the table below and selecting “Update Field”. This feature is located in the “Captions” section of the “References” tab in MS Word.* ***Note: Remove this instructional paragraph.***

[Figure 2 - Sample Use Case Diagram from Bruegge & DuToit ( modified ) **Error! Bookmark not defined.**](#_Toc525544068)

[Figure 3 - Sample Use Case Diagram from Robertson and Robertson **Error! Bookmark not defined.**](#_Toc525544069)

*On a related note, the references in the paragraph above, “( see* ***Error! Reference source not found.*** *on page* ***Error! Bookmark not defined.*** *)” include cross-references to the Figure and page number that will adjust automatically when other Figures or pages are added or removed. This is done with the “cross-reference” button in the “Captions” section of the “References” tab in MS Word.* ***Note: Remove this instructional paragraph.***

### ****List of Tables****

*If a document contains a large number of tables, then it is appropriate to include a list of tables at the beginning of the document, following the table of contents. Each table should include a title, and be numbered in a consistent logical fashion. The following list of tables was automatically generated from table captions ( see below), and can be automatically updated by right-clicking on the table below and selecting “Update Field”. This feature is located in the “Captions” section of the “References” tab in MS Word.* ***Note: Remove this instructional paragraph.***

[*Table 2 - Requirements - Acceptance Tests Correspondence* **Error! Bookmark not defined.**](#_Toc525544085)

# Project Description

*Short Version ( SV ): Section I of the document provides sufficient information about the project and its domain to put the materials being tested and inspected into proper context.*

## Project Overview

*SV: Provide a brief quick description of the project, sufficient to give the reader a general understanding of what the project is all about. Refer to other documents for further details if appropriate. ( See also section 3 below. )*

*( e.g. the project description and specification documents, requirements documents, design documents, etc. )*

Your text goes here . . .

## Project Domain

*SV: In this section provide relevant information regarding the domain of the project, particularly as necessary to fully understand the material being tested and to evaluate its value and/or correctness.*

Your text goes here . . .

## Relationship to Other Documents

*SV: Document here the relationship between this document and other documents related to the project, such as the Project Description Document, Project Requirements Document, Project Design Document, etc. Make sure to include all relevant documents in the bibliography, and to reference them as appropriate here.*

Your text goes here . . .

## Naming Conventions and Definitions

*SV: Define terminology to avoid miscommunications or misunderstandings.*

*This section can be very short or even omitted if referring to a previous document in which ( much of ) this information is provided. Or this section could provide additional information not documented well elsewhere. In particular, only information needed to understand and appreciate the testing and inspection processes need be included.*

*Note, however, that this document should be understandable as a stand-alone document, without having to go look up things in other documents. Any terminology that is needed to understand this report should be included here.*

### Definitions of Key Terms

*SV: Define* ***words*** *that may have special or multiple meanings.*

*All Terms, Including Acronyms and Abbreviations, Used in the Project must be defined at some point. List the most important ones here, and refer the reader to the glossary on page 17 for a complete list. ( Note: that page number is a cross-reference, and will automatically be updated whenever the glossary moves. )*

*Content*

*A glossary containing the meanings of all names, acronyms, and abbreviations used within the requirements specification. Select names carefully to avoid giving a different, unintended meaning.*

*This glossary reflects the terminology in current use within the work area. You might also build on the standard names used within your industry.*

*For each term, write a succinct definition. The appropriate stakeholders must agree on this definition.*

*Avoid abbreviations, as they introduce ambiguity, require additional translations, and could potentially lead to misinterpretation in the mind of anyone who is trying to understand your requirements. Ask your requirements analysts to replace all abbreviations with the correct term. This is easily done with word processors.*

*Acronyms are acceptable if they are completely explained by a definition.*

*Motivation*

*Names are very important. They invoke meanings that, if carefully defined, can save hours of explanations. Attention to names at this stage of the project helps to highlight misunderstandings.*

*The glossary produced during requirements is used and extended throughout the project.*

*Examples*

*Truck: A vehicle used for spreading de-icing material on roads. “Truck” is not used to refer to goods-carrying vehicles.*

*BIS: Business Intelligence Service. The department run by Steven Peters to supply business intelligence for the rest of the organization.*

*Considerations*

*Make use of existing references and data dictionaries. Obviously, it is best to avoid renaming existing items unless they are so ambiguous that they cause confusion.*

*From the beginning of the project, emphasize the need to avoid homonyms and synonyms. Explain how they increase the cost of the project.*

Your text goes here . . .

### UML and Other Notation Used in This Document

*SV: Define* ***symbols****,* ***diagrams****, and other* ***notations*** *used. May refer to a standard reference, such as “UML Distilled” by Fowler. ( Include in bibliography. )*

*Content*

*This section should describe the specific meaning of any symbols, punctuation, subscripts, superscripts, etc. used commonly throughout the document. If following published or common standards, then it is acceptable to reference those standards, and list any exceptions.*

*Motivation*

*If the distinction between a hollow arrow and a solid arrow is significant, for example, then everyone must know exactly what the distinctions and meanings are.*

*Considerations*

*If a particular notation is only used in one place, say on a single diagram or in a single section, then it may be more appropriate to document it in that specific location.*

*Example*

*This document generally follows the Version 2.0 OMG UML standard, as described by Fowler in [4]. Any exceptions are noted where used.*

Your text goes here . . .

### Data Dictionary for Any Included Models

*SV: Define* ***data structures*** *and* ***data properties*** *relative to this project, such as the contents of an employee record or the fact that student GPA ranges from 0.0 to 4.0 corresponding to letter grades of F to A. Data file formats may be referenced to documented standards, such as jpg or pdf.*

*Content*

*Dictionary definitions of all information flows and stores used in models. Particular consideration should be given to defining the data attributes of all flows shown the context models (see sections 7 and 8).*

*This section should also contain any technical specifications for interfaces shown on the context models.*

*Motivation*

*The context diagram provides an accurate definition of the scope of the work being studied or the scope of the product to be built. This definition can be completely accurate only if the information flows bordering the scope have their attributes defined.*

*Examples*

*Road de-icing schedule = issue number + {road section identifier + treatment start time + critical start time + truck identifier} + depot identifier*

*As you progress through the requirements specification, define each of the elementary terms in detail.*

*Considerations*

*The dictionary provides a link between the requirements analysts and the implementers. The implementers add implementation details to the terms in the dictionary, defining how the data will be implemented. Also, implementers add terms that are present because of the chosen technology and that are independent of the business requirements.*

Your text goes here . . .

# Project Deliverables

*SV: This portion of the report documents what your group has produced this semester.*

Your text goes here . . .

## First Release

*SV: Document the date of the first release and the functionality of the system as of that point. UML diagrams and/or screenshots may be appropriate.*

Your text goes here . . .

## Second Release

*SV: Document the date of the second release and the functionality of the system as of that point. UML diagrams and/or screenshots may be appropriate.*

Your text goes here . . .

## Comparison with Original Project Design Document

*SV: Document how the prototype produced compares with the full project described by the previous group. Don’t forget to reference their work properly.*

Your text goes here . . .

# Testing

*SV: This portion of the report documents the testing procedures applied and the results obtained. For CS 440 each student is to contribute a significant piece of code to be tested. ( It is usually best to have code tested by someone other than the person who wrote it. For CS 440 a round-robin approach may work well, with student A testing student B’s code, student B testing student C’s code, etc. )*

## Items to be Tested

*SV: Document here the items to be tested. Subsections may be needed. Provide context if necessary, e.g. an overall UML diagram showing where the items to be tested fit into the overall project. Providing item ID numbers here will make later sections easier.*

## Test Specifications

*SV: Document here each of the tests to be run. For each test, provide the following, ( as applicable. )*

**ID# - Name**

**Description:** Your description here . . . *Generally a single line.*

**Items covered by this test:** Your list here . . .

**Requirements addressed by this test:** *If applicable. Simple requirement numbers as referred to in the requirements document ( or requirements section of the project description document for CS 440. )* Your list here . . .

**Environmental needs:** *As needed, list any special HW, SW, or other resources needed to run this test. This may include stubs, drivers, and any other needed scaffolding.*Your list here . . .

**Intercase Dependencies:** *If applicable, list any other tests that must be completed ( successfully ) before this test can be conducted. Other tests that depend on this one should also be documented here. Else “NA”.* Your list here . . .

**Test Procedures:** *The steps need to be taken to run the test. If a ( automated ) test script is provided, then this may be as simple as running the script.* Your list here . . .

**Input Specification:** *The input values used to run this test. Depending on the specific test case(s), this may take the form of specific values, a description, ( e.g. “three positive integers, randomly chosen” ), a table, or the name of a data file containing the necessary input.* Your list here . . .

**Output Specifications:** *The output results expected from running this test.**Depending on the specific test case(s), this may take the form of specific values, a description, ( e.g. “the positive square root of the input value provided” ), a table, or the name of a data file containing the expected results for comparison purposes ( e.g. diff. )* Your list here . . .

***Note: In some cases it may be appropriate to combine input and output specifications into a single combined table, with rows for different test cases.***

**Pass/Fail Criteria:** *A.k.a. acceptance criteria, this documents what it means to “pass” the test.* Your criteria here . . .

## Test Results

*SV: Document here the result of running each test. Include the items addressed by each test, the time(s) and date(s) the tests were run, who ran the tests, the success or failure status of the test, and any faults identified as a result of running the tests.*

**ID# - Name**

**Date(s) of Execution:** *Provide times also only if relevant, i.e. if the test(s) were run multiple times on the same date with different results.* Your data here . . .

**Staff conducting tests:** *Who conducted the test(s)?*Your list here . . .

**Expected Results:** Your list here . . .

**Actual Results:** Your list here . . .

**Test Status:** *Generally “Pass” or “Fail” Add explanatory notes if appropriate.* Your status here . . .

## Regression Testing

*SV: Document any tests that are repeated, whether they passed or failed the initial tests. ( This section would particularly apply to older tests that must be repeated as a result of running the current tests, or to document the conditions under which it will be necessary to re-run these tests in the future. As such it may not apply to CS 440. )*

# Inspection

## Items to be Inspected

* + Trader Class
  + Location Data Class
  + Map Movement Logic Classes
  + Silk Road Game Class

## Inspection Procedures

Inspection Checklist -

1. Class Scope
2. Unit Test Coverage
3. Library Usage – Security and Performance

Inspections are to be done in a mix of during meetings and in person. The primary focus is on members of the team to inspect their classes and have the results be brought up in meetings for any questions with other members and discussions on design choices. 2 meetings were held in this manner. The

## Inspection Results

|  |  |
| --- | --- |
| Class Scope | All tested classes were sufficiently encapsulated in their scope and didn’t have any overreach. |
| Unit Test Coverage | Trader Class and Location Data Class have sufficient unit test coverage. Map Movement Class and Silk Road Game Class have some coverage. |
| Library Usage – Security and Performance | Libraries have been included through which allows the team to update libraries easily.   * Json-simple library hasn’t been updated in a almost a decade. This can present in security issues and can be relatively slow to newer JSON parsing libraries. * JUNIT library version is outdated. * OpenJFX library version is outdated. |

## Recommendations and Conclusions

Unit test coverage for some classes such as Map Movement Logic Classes and Silk Road Game Class were found to be lacking and certain sections should be abstracted away to facilitate this. The json-simple library used in Location Data class is concerning for security reasons as the library hasn’t been updated in almost a decade and JSON parsers have been shown to be vulnerable to various attacks. There needs to be more care in keeping libraries up to date and dealing with possible regressions because of this.

# Project Issues

*These sections were copied from the development project template, and may or may not apply to the coding project. However do please complete the project retrospective in any case.*

## Open Issues

*SV: Issues that have been raised and do not yet have a conclusion.*

*Content*

*A statement of factors that are uncertain and might make significant difference to the product.*

*Motivation*

*To bring uncertainty out in the open and provide objective input to risk analysis.*

*Examples*

*Our investigation into whether the new version of the processor will be suitable for our application is not yet complete.*

*The government is planning to change the rules about who is responsible for gritting the motorways, but we do not know what those changes might be.*

*Considerations*

*Are there any issues that have come up from the requirements gathering that have not yet been resolved? Have you heard of any changes that might occur in the other organizations or systems on your context diagram? Are there any legislative changes that might affect your system? Are there any rumors about your hardware or software suppliers that might have an impact?*

Your text goes here . . .

## Waiting Room

*SV: This is a place to record ideas or wishes that will not be included in the current release of the product, but which might be worth reconsidering at a later date.*

*Requirements that will not be part of the next release. These requirements might be included in future releases of the product.*

*Content*

*Any type of requirement.*

*Motivation*

*To allow requirements to be gathered, even though they cannot be part of the current development. To ensure that good ideas are not lost.*

*Considerations*

*The requirements-gathering process often throws up requirements that are beyond the sophistication of, or time allowed for, the current release of the product. This section holds these requirements in waiting. The intention is to avoid stifling the creativity of your users and clients, by using a repository to retain future requirements. You are also managing expectations by making it clear that you take these requirements seriously, although they will not be part of the agreed-upon product.*

*Many people use the waiting room as a way of planning future versions of the product. Each requirement in the waiting room is tagged with its intended version number. As a requirement progresses closer to implementation, then you can spend more time on it and add details such as the cost and benefit attached to that requirement.*

*You might also prioritize the contents of your waiting room. “Low-hanging fruit”—requirements that provide a high benefit at a low cost of implementation—are the highest-ranking candidates for the next release. You would also give a high waiting room rank to requirements for which there is a pent-up demand.*

Your text goes here . . .

## Ideas for Solutions

*SV: When developing requirements only, it is not the role of the business analyst to dictate the implementation of the solution.  However they can pass along any ideas they have here as suggestions to the developers.  For CS 440 this report includes system and object design, so this section would make suggestions for implementation and testing that would come after design, such as the use of a particular language, IDE, library, or other tools.*

*When you gather requirements, you focus on finding out what the real requirements are and try to avoid coming up with solutions. However, when creative people start to think about a problem, they always generate ideas about potential solutions. This section of the template is a place to put those ideas so that you do not forget them and so that you can separate them from the real business requirements.*

*Content*

*Any idea for a solution that you think is worth keeping for future consideration. This can take the form of rough notes, sketches, pointers to other documents, pointers to people, pointers to existing products, and so on. The aim is to capture, with the least amount of effort, an idea that you can return to later.*

*Motivation*

*To make sure that good ideas are not lost. To help you separate requirements from solutions.*

*Considerations*

*While you are gathering requirements, you will inevitably have solution ideas; this section offers a way to capture them. Bear in mind that this section will not necessarily be included in every document that you publish.*

Your text goes here . . .

## Project Retrospective

*Content*

We were able to conduct meetings at least once a week or more. These meetings were invaluable in planning our sprints, checking our progress, and for guiding each other. There were some difficulties in the beginning with setting up each team member’s environment regarding getting the right JavaFX version and Java IDE. It was more work to guide each other asynchronously for these issues.

*Considerations*

There were certain bottlenecks in development, especially in the early stages. There were points where one part of the project was necessary to be completed in order for other team members to progress. As a result of this, there were some slowdowns while things like map movement was defined. This didn’t mesh well with JIRA as tickets were stuck in progress till completion.

# Glossary

*SV: The glossary is a more complete and inclusive dictionary of defined terms than that found in section I.7.a, the latter of which only covered the most important key terms needed to understand the report.*

*The glossary defines terms that may not be familiar to all readers. This is especially important if the document is expected to reach a wide and varied audience, such as school children. The glossary may be placed at either the beginning or the end of the document.*

***Flotsam:*** *Any part of a ship or its cargo found floating on the water, whether it was deliberately or accidentally lost by its original owners.*

***Jetsam:*** *Any part of a ship or its cargo that is deliberately cast off ( jettisoned ) by its original owners, generally in order to lighten the ship, whether it floats or sinks.*

Your text goes here . . .

# References / Bibliography

*This section describes the documents and other sources from which information was gathered. This sample bibliography was generated using the “Insert Citation” and “Bibliography” buttons in the “Citations & Bibliography” section under the “References” tab of MS Word. Creating new citations will not update this list unless you click on it and select “Update Field”. You may need to reset the style for this paragraph to “normal” after updating.*

***DO NOT NEGLECT TO REFERENCE THE PREVIOUS GROUP’S REPORT ! ! !***

|  |  |
| --- | --- |
| **[1]** | **Robertson and Robertson, Mastering the Requirements Process.** |
| **[2]** | **A. Silberschatz, P. B. Galvin and G. Gagne, Operating System Concepts, Ninth ed., Wiley, 2013.** |
| **[3]** | **J. Bell, "Underwater Archaeological Survey Report Template: A Sample Document for Generating Consistent Professional Reports," Underwater Archaeological Society of Chicago, Chicago, 2012.** |
| **[4]** | **M. Fowler, UML Distilled, Third Edition, Boston: Pearson Education, 2004.** |

# Index

*This section provides an index to the report. The sample below was generated using the “Mark Entry” and “Insert Index” items from the “Index” section on the “References” tab, and can be automatically updated by right clicking on the table below and selecting “Update Field”. To remove marked entries from the document, toggle the display of hidden paragraph marks ( the paragraph button on the “Home” tab ), and remove the tags shown with XE in { curly braces. }*

**No index entries found.**