



**<Project Name>
Supplementary Specification**

Version <1.0>

[Note: The following template is provided for use with the Rational Unified Process. Text enclosed in square brackets and displayed in blue italics (style=InfoBlue) is included to provide guidance to the author and should be deleted before publishing the document. A paragraph entered following this style will automatically be set to normal (style=Body Text).]

[To customize automatic fields in Microsoft Word (which display a gray background when selected), select File>Properties and replace the Title, Subject and Company fields with the appropriate information for this document. After closing the dialog, automatic fields may be updated throughout the document by selecting Edit>Select All (or Ctrl-A) and pressing F9, or simply click on the field and press F9. This must be done separately for Headers and Footers. Alt-F9 will toggle between displaying the field names and the field contents. See Word help for more information on working with fields.]

<Project Name>	Version: <1.0>
Supplementary Specification	Date: <dd/mmm/yy>
<document identifier>	

Revision History

Date	Version	Description	Author
<dd/mmm/yy>	<x.x>	<details>	<name>

CTOSystem.com

<Project Name>	Version: <1.0>
Supplementary Specification	Date: <dd/mmm/yy>
<document identifier>	

Table of Contents

1.	Introduction	4
1.1	Purpose	4
1.2	Scope	4
1.3	Definitions, Acronyms and Abbreviations	4
1.4	References	4
1.5	Overview	4
2.	Functionality	4
2.1	<Functional Requirement One>	5
3.	Usability	5
3.1	<Usability Requirement One>	5
4.	Reliability	5
4.1	<Reliability Requirement One>	5
5.	Performance	5
5.1	<Performance Requirement One>	6
6.	Supportability	6
6.1	<Supportability Requirement One>	6
7.	Design Constraints	6
7.1	<Design Constraint One>	6
8.	Online User Documentation and Help System Requirements	6
9.	Purchased Components	6
10.	Interfaces	6
10.1	User Interfaces	6
10.2	Hardware Interfaces	6
10.3	Software Interfaces	6
10.4	Communications Interfaces	6
11.	Licensing Requirements	7
12.	Legal, Copyright and Other Notices	7
13.	Applicable Standards	7

<Project Name>	Version: <1.0>
Supplementary Specification	Date: <dd/mmm/yy>
<document identifier>	

Supplementary Specification

1. Introduction

*[The introduction of the **Supplementary Specification** should provide an overview of the entire document. It should include the purpose, scope, definitions, acronyms, abbreviations, references, and overview of this **Supplementary Specification**.*

*The **Supplementary Specification** captures the system requirements that are not readily captured in the use cases of the use-case model. Such requirements include:*

- *Legal and regulatory requirements, including application standards.*
- *Quality attributes of the system to be built, including usability, reliability, performance, and supportability requirements.*
- *Other requirements such as operating systems and environments, compatibility requirements, and design constraints.]*

1.1 Purpose

*[Specify the purpose of this **Supplementary Specification**.]*

1.2 Scope

*[A brief description of the scope of this **Supplementary Specification**; what Project(s) it is associated with, and anything else that is affected or influenced by this document.]*

1.3 Definitions, Acronyms and Abbreviations

*[This subsection should provide the definitions of all terms, acronyms, and abbreviations required to properly interpret the **Supplementary Specification**. This information may be provided by reference to the project Glossary.]*

1.4 References

*[This subsection should provide a complete list of all documents referenced elsewhere in the **Supplementary Specification**. Each document should be identified by title, report number (if applicable), date, and publishing organization. Specify the sources from which the references can be obtained. This information may be provided by reference to an appendix or to another document.]*

1.5 Overview

*[This subsection should describe what the rest of the **Supplementary Specification** contains and explain how the document is organized.]*

2. Functionality

[This section describes the functional requirements of the system for those requirements which are expressed in the natural language style. For many applications, this may constitute the bulk of the SRS Package and thought should be given to the organization of this section. This section is typically organized by feature, but alternative organization methods, for example organization by user or organization by subsystem, may also be appropriate. Functional requirements may include feature sets, capabilities, and security.

Where application development tools, such as requirements tools, modeling tools, etc., are employed to capture the functionality, this section document will refer to the availability of that data, indicating the location and name of the tool used to capture the data.]

<Project Name>	Version: <1.0>
Supplementary Specification	Date: <dd/mmm/yy>
<document identifier>	

2.1 <Functional Requirement One>

[The requirement description.]

3. Usability

[This section should include all of those requirements that affect usability. Examples follow:

- *specify the required training time for a normal users and power users to become productive at particular operations*
- *specify measurable task times for typical tasks, or*
- *specify requirements to conform to common usability standards, for example, IBM's CUA standards or Microsoft's GUI standards]*

3.1 <Usability Requirement One>

The requirement description.

4. Reliability

[Requirements for reliability of the system should be specified here. Suggestions are as follows:

- *Availability – specify percentage of time available (xx.xx%), hours of use, maintenance access, degraded mode operations, etc.*
- *Mean Time Between Failures (MTBF) – this is usually specified in hours but it could also be specified in terms of days, months or years.*
- *Mean Time To Repair (MTTR) – how long is the system allowed to be out of operation after it has failed?*
- *Accuracy – specify precision (resolution) and accuracy (by some known standard) that is required in the systems output.*
- *Maximum bugs or defect rate – usually expressed in terms of bugs/KLOC (thousands of lines of code), or bugs/function-point.*
- *Bugs or defect rate – categorized in terms of minor, significant, and critical bugs: the requirement(s) must define what is meant by a “critical” bug (e.g., complete loss of data or complete inability to use certain parts of the functionality of the system).]*

4.1 <Reliability Requirement One>

[The requirement description.]

5. Performance

[The performance characteristics of the system should be outlined in this section. Include specific response times. Where applicable, reference related Use Cases by name.

- *Response time for a transaction(average, maximum)*
- *Throughput (e.g., transactions per second)*
- *Capacity (e.g., the number of customers or transactions the system can accommodate)*
- *Degradation modes (what is the acceptable mode of operation when the system has been degraded in some manner)*
- *Resource utilization: memory, disk, communications, etc.]*

<Project Name>	Version: <1.0>
Supplementary Specification	Date: <dd/mmm/yy>
<document identifier>	

5.1 <Performance Requirement One>

[The requirement description.]

6. Supportability

[This section indicates any requirements that will enhance the supportability or maintainability of the system being built, including coding standards, naming conventions, class libraries, maintenance access, maintenance utilities.]

6.1 <Supportability Requirement One>

[The requirement description.]

7. Design Constraints

[This section should indicate any design constraints on the system being built. Design constraints represent design decisions that have been mandated and must be adhered to. Examples include software languages, software process requirements, prescribed use of developmental tools, architectural and design constraints, purchased components, class libraries, etc.]

7.1 <Design Constraint One>

[The requirement description.]

8. Online User Documentation and Help System Requirements

[Describes the requirements, if any, for on-line user documentation, help systems, help about notices, etc.]

9. Purchased Components

[This section describes any purchased components to be used with the system, any applicable licensing or usage restrictions, and any associated compatibility/interoperability or interface standards.]

10. Interfaces

[This section defines the interfaces that must be supported by the application. It should contain adequate specificity, protocols, ports and logical addresses, etc., so that the software can be developed and verified against the interface requirements.]

10.1 User Interfaces

[Describe the user interfaces that are to be implemented by the software.]

10.2 Hardware Interfaces

[This section defines any hardware interfaces that are to be supported by the software, including logical structure, physical addresses, expected behavior, etc.]

10.3 Software Interfaces

[This section describes software interfaces to other components of the software system. These may be purchased components, components reused from another application or components being developed for subsystems outside of the scope of this SRS, but with which this software application must interact.]

10.4 Communications Interfaces

[Describe any communications interfaces to other systems or devices such as local area networks, remote serial devices, etc.]

<Project Name>	Version: <1.0>
Supplementary Specification	Date: <dd/mmm/yy>
<document identifier>	

11. Licensing Requirements

[Defines any licensing enforcement requirements or other usage restriction requirements which are to be exhibited by the software.]

12. Legal, Copyright and Other Notices

[This section describes any necessary legal disclaimers, warranties, copyright notices, patent notice, wordmark, trademark or logo compliance issues for the software.]

13. Applicable Standards

[This section describes by reference any applicable standards and the specific sections of any such standards that apply to the system being described. For example, this could include legal, quality and regulatory standards, industry standards for usability, interoperability, internationalization, operating system compliance, etc.]