# SOFTWARE REQUIREMENTS SPECIFICATION
# Outline and Requirements

***General comments**: The following guidelines are given in italics and represent the bare minimum required to complete this plan. Non-italicized text is to be included in your plan verbatim. All sections must be included in your plan, even if you do not feel it applies.*

## 1. Introduction

*One or two paragraphs which introduce the document.*

### 1.1. System reference

*One or two paragraphs which briefly describes the overall system within which the software product executes. This description identifies the context for the software product.*

### 1.2. Overall description

*One or two paragraphs which briefly describes the software product. This description is an overview of the product itself which introduces it to the reader.*

### 1.3. Software project constraints

*One or two paragraphs which briefly identifies the major constraints imposed upon the product. Use a list format if there are more than two constraints.*

## 2. Information Description

### 2.1. Information flow representation

*One or two paragraphs which introduce the major information flow of the software product. Also identify nomenclature and symbols to be used in this section.*

#### 2.1.1. Data flow

*Using a hierarchical decomposition of data flow diagrams, show, in detail, the data flow of the software product and all its subsystems. Clearly describe the dataflow diagrams. Use consistent naming conventions for all data. This section is usually quite lengthy and care must be taken to organize the presentation well.*

#### 2.1.2. Control flow

*Using finite state machine models or other similar methods, show, in detail, the flow of control in the software product and all its subsystems. Clearly describe all diagrams. Ensure all names are consistent with the data flow diagrams in the previous section. This section is usually quite lengthy and care must be taken to organize the presentation well.*

### 2.2. Information content representation

*A complete description of all data identified in the Information flow representation section above is given in this section. The data type, size, description, and other attributes of each named data item is given, usually in tabular form.*

### 2.3. Standard interface description

*Clearly describe the hardware, software, and human interfaces to external system elements and internal software functions. Use drawings as necessary to show where the interfaces exist and the contents of the interfaces.*

## 3. Functional Description

*Give one or two paragraphs which summarizes the functional description of the software product.*

### 3.1. Functional partitioning

*Provide a partitioned functional representation of the software product in a hierarchical fashion. Use a formal representation, use text to clarify the figures, carry the decomposition to the module unit level.. Ensure all named data or control items are consistent with the data and control-flow models described above.*

### 3.2. Functional description

*For each module identified above:*

#### 3.2.1. Processing narrative

*Describe the operation of the module.*

#### 3.2.2. Restrictions and limitations

*Identify all restrictions and limitations imposed upon the module.*

#### 3.2.3. Performance requirements

*Identify all timing, delay, or other performance constraints for the module. Note that module performance requirements are usually derived from system and subsystem constraints given in the Functional Partitioning section above. For time constrained software, a flowdown of module delays is documented either in this section or in the Functional Partitioning section above.*

#### 3.2.4. Design constraints

*Identify and justify all design constraints for each module.*

#### 3.2.5. Supporting diagrams

*Show and describe all data-flow and control flow diagrams for the module.*

### 3.3. Control description

*For each module:*

#### 3.3.1. Control specification

*Identify the control inputs and outputs for the module.*

#### 3.3.2. Design constraints

*Identify all design constraints for the module.*

## 4. Behavioral description

### 4.1. System states

*Decompose, show, and describe all finite state diagrams and/or decision charts reflecting the control of the system, subsystems, and significant control-intensive modules. Clearly describe the inputs, output ID's, and states for all control state transitions.*

### 4.2. Events and actions

*Clearly identify and describe all events creating the inputs, actions performed by the output ID's.*

## 5. Validation criteria

### 5.1. Performance bounds

*Identify the performance expectations and bounds for the software product and its major subsystems. Testing will validate against these bounds.*

### 5.2. Classes of tests

*Give a table and description of all major classes fo tests to be applied to the software once constructed to assure compliance with requirements specs.*

### 5.3. Expected software response

*For each test class, define the expected software response.*

### 5.4. Special considerations

*Claify any special issues relating to validation testing such as special setup conditions, additional resources necessary to conduct the tests, etc.*

## 6. Bibliography

*Give references to all documentation used to create this specification including:*

- *other software engineering documentation*
- *technical references,*
- *vendor literature,*
- *standards*

## 7. Appendix