Software Specifications

**Requirements Specifications:** state the desired functional and performance characteristics of some component independent of any actual realization.

**Design Specifications:** describe the component’s real internal structure and behavior.
Software Requirement Engineering

- Requirements analysis

- Requirements documentation

Sep.19, 2000
Requirement Analysis

**Partitioning:** decomposing a system into a hierarchy of elements.

**Allocation:** each system level requirement is allocated to one or more elements at the next level.

**Flowdown:** consists of writing requirements for the lower level elements in response to the allocation. When a system requirement is allocated to a subsystem, the subsystem must have at least one requirement that responds to the allocation.
## Allocation

<table>
<thead>
<tr>
<th>System Requirements</th>
<th>Subsystem A</th>
<th>Subsystem B</th>
<th>Subsystem C</th>
</tr>
</thead>
<tbody>
<tr>
<td>sys 001</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>sys 002</td>
<td>X</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>sys 003</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>sys 004</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>sys 005</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>sys 006</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>sys 007</td>
<td>X</td>
<td></td>
<td>X</td>
</tr>
</tbody>
</table>
## Flowdown

<table>
<thead>
<tr>
<th>System Requirement</th>
<th>system A Requirement</th>
<th>System B Requirement</th>
<th>System C Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYS 001</td>
<td>SSA 001, SSA 002</td>
<td>SSB 001</td>
<td></td>
</tr>
<tr>
<td>SYS 002</td>
<td>SSA 003, SSA 004, SSA 005</td>
<td></td>
<td>SSC 001, SSC 002</td>
</tr>
<tr>
<td>SYS 003</td>
<td></td>
<td>SSB 002, SSB 003</td>
<td></td>
</tr>
<tr>
<td>SYS 004</td>
<td>SSA 006, SSA 007</td>
<td>SSB 004, SSB 005, SSB 006</td>
<td>SSC 003</td>
</tr>
<tr>
<td>SYS 005</td>
<td></td>
<td></td>
<td>SSC 004, SSC 005</td>
</tr>
<tr>
<td>SYS 006</td>
<td></td>
<td>SSB 007, SSB 008</td>
<td></td>
</tr>
<tr>
<td>SYS 007</td>
<td>SSA 008, SSA 009</td>
<td>SSB 009</td>
<td></td>
</tr>
</tbody>
</table>
Requirements Analysis Methods

Process-oriented methods take the primary viewpoint of the way the system transforms inputs into outputs.

Data-oriented methods emphasize the system state as a data structure.

Control-oriented methods emphasize synchronization, deadlock, execution, concurrency, and process activation and deactivation.

Object-oriented methods base analysis on classes of objects of the system and their interactions.
Software Requirements Specifications

1. Unambiguous
2. Complete
3. Verifiable
4. Consistent
5. Modifiable
6. Traceable
7. Usable during the Operational and Maintenance Phase
Prototype SRS Outline

Table of Contents

1. Introduction
   (a) Purpose
   (b) Scope
   (c) Definitions, Acronyms, and Abbreviations
   (d) Overview

2. General Description
   (a) Product Perspective
   (b) Product Functions
   (c) User Characteristics
   (d) General Constraints
   (e) Assumptions and Dependencies

3. Specific Requirements
   (a) Functional Requirements
(b) External Interface Requirements

(c) Performance Requirements

(d) Design Constraints

(e) Attributes

(f) other Requirements

Appendixes

Index
Requirements Analysis Methods

**Process-oriented methods** take the primary viewpoint of the way the system transforms inputs into outputs.

**Data-oriented methods** emphasize the system state as a data structure.

**Control-oriented methods** emphasize synchronization, deadlock, execution, concurrency, and process activation and deactivation.

**Object-oriented methods** base analysis on classes of objects of the system and their interactions.
Structured Analysis

1. major constructs generic to the problem being modeled.

2. relationships that associate any one construct with others in the model.

3. properties that either define or describe an object type.
Structured Analysis

- Process
- Data Flow
- Data Store
- External Entity
- Data Group
- Data Element
Structured Analysis

- Data Flow Diagram
- Data Dictionary
- The Primitive Process Specifications
- Structure Walk through
**Entity-Relationship Model**

**entities:** an entity stands for a collection of item that share common properties.

**relation:** the relations are in which an entity participates. degree of relationship: is the number of entities associated in the relationship.

- Recursive
- Binary
- Ternary
- n-ary relationship

**attributes:** the properties of an entity are its attributes.
Finite State Machine

- a finite set of states, $Q$;
- a finite set of inputs, $I$;
- a transition function: $Q \times I \rightarrow Q$.

- State Transition Diagram
- State Transition Matrices
Petri Nets

- a finite set of places;

- a finite set of transitions;

- a finite set of arrows connecting either places to transitions or transitions to places.
Petri Nets

- A marking of a PN consists of assigning a nonnegative integer to each place.

- A transition is said to be enabled if there is at least one token in each of its input places.

- A fire means that one token is removed from each input place and one token is inserted in each output place of the transition.