Requirements Specification

Abstract

- Summarizes in a few sentences what the design is

System Description

- Gives an overview and functional description of the system
- Written from a client's point of view

External view of the system

- All system inputs and outputs are identified and briefly described

- Use Case Diagram
  - Specifies the graphical and textual view of the use cases and the actors
  - Each use case must include
    - A textual description of the use case including
      - Responsibilities and operation
      - Exceptions to the use case

Functional Behavior

- Gives a description of the required behavior of the system from an external point of view
- Considerations
  - How to start the system
  - How to stop it
  - Requirements for use
    - Pre conditions
    - Post conditions
    - Rules for use
  - Constraints

Environmental Considerations and Requirements

- Speed
- Memory requirements
- Host requirements
- Display requirements

Design and Design Documentation

Abstract

- Summarizes in a few sentences what the design is and does

System Description

- Gives a detailed functional description of the system
- Written from a designer’s point of view

Functional Decomposition

- Identify
  - Major software functional blocks of the design
  - Information that must flow between the functional blocks

CRC Cards

- Map the functionality expressed in functional decomposition to
  - Classes
  - Responsibilities of each class
- Map the information flow between functional blocks
  - Collaborators with each class
- Identify
  - Superclass(es)
  - Subclass(es)

Classes and Functions

- Classes
  - Identify the classes
  - Brief textual description for each class
    - Initialization requirements
    - Invariants
    - Access specifications(s)
- Functions
  - Identify the function
  - Brief textual description for each function
  - Identify
    - Pre conditions
    - Invariants
    - Post conditions

UML Class Diagram

- For each class develop a class diagram
  - Data members
  - Function members
- Structure
  - Identify relationships between or among classes

- Inheritance
- Composition
- Uses
- Creates

Test Plan

- Based upon Requirements Specification
- For each use case
  - Identify what must be tested to verify the use case

Test Cases

- Based upon the Test Plan and the Requirements Specification
- For each use case
  - Identify how the design must be tested to verify the use case
  - The test code implementation to execute the test
  - Identify
    - What is expected when the test is executed
    - The test result