

Software Requirements

Definition : Description of something the system is capable of doing in order to fulfill the system's purpose

Agenda

- Definitions
- Characteristics of requirements.
- Types of requirements
- Examples:
 - Traffic Violation Reports System

Definitions

- Requirement – A feature/property of the system or a description of something the system is capable of doing in order to fulfill the system's purpose.
- Software Requirements document:
 - A set of precisely stated requirements that the software must satisfy.
 - Establishes boundaries on the solution space of the problem of developing a useful software system.
 - Allows a design to be validated – if the constraints and properties specified in the document are satisfied by the software design, then that design is an acceptable solution to the problem.
 - A “contract” between developers and clients.

Agenda

- Definitions
- Characteristics of requirements.
- Types of requirements
- Examples:
 - Traffic Violation Reports System

Characteristics of requirements

- Correct
- Unambiguous
- Complete
- Testable
- Consistent
- Deal only with the problem
- Modifiable
- Traceable

Correct

- Each requirement statement accurately represents the functionality required of the system to be built.
- Example (of an incorrect requirement):
 - Problem domain (real life) states that policeman ID numbers are in the range [10000...) and the requirements document specifies that each policeman has an ID number.

Unambiguous

- The difficulty of ambiguity stems from the use of natural language which in itself is inherently ambiguous.
- There is one and only one interpretation for every requirement.
- Requirement statements should be short, explicit, precise and clear.
- A glossary should be used when a term used in a particular context could have multiple meanings (I.e. “the user”).
- Formal requirements languages help reduce ambiguity.

Unambiguous (cont.)

- Examples (of ambiguity):
 - The TVRS shall store changes made in the details of a traffic report *as soon as the data is entered.*
- Disambiguation:
 - The TVRS shall store changes made in the details of a traffic report if and only if all input fields are valid and user approved saving of data (see sections ...)

Complete

- A requirements document is complete if it includes all of the significant requirements, whether relating to functionality, performance, design constraints attributes or external interfaces.
- No sections are marked “To be determined” (TBD).
- Conforms to the company standards.

Testable

- A requirements document is testable (verifiable) if and only if every requirement statement in it is testable.
- A requirement is testable if and only if there is some finite cost-effective way in which a person or machine can check to see if the software product satisfies that requirement.

Testable (cont.)

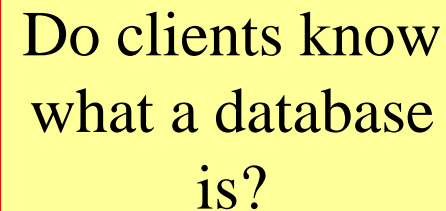
- Example of a non-verifiable requirement:
 - The TVRS shall complete storage of data within a reasonable time of the user confirming a “Save” sequence.
- Example of a verifiable requirement:
 - The TVRS shall complete storage of data within 5 seconds of the user confirming a “Save” sequence, 80% of the time.

Consistent

- Three types of conflicts:
 - Different terms used for the same object:
 - F323 and a “Policeman Details Form” might be used to describe the same form.
 - Characteristics of objects:
 - In one part of the requirements document: “A policeman ID shall consist of *decimal digits only*”, while in another part “Incase the policeman ID consists of *non-alphanumerical characters*, display an error message”.

Consistent (cont.)

- Logical or temporal faults: “A follows B” in one part, “A and B occur simultaneously” in another.
 - “TVRS shall support removal of a policeman record from the personal *database*” vs. “TVRS shall support read-only access to policeman details”.



Do clients know
what a database
is?

Deal only with the problem

- Requirements should state “what” is required at the appropriate system level, not “how”.
 - In some cases, a requirement may dictate how a task is to be accomplished.
 - Normally, you try to specify functionality rather than construction details.
- Avoid telling the designer “how” to do this job, instead state “what” has to be accomplished.
- Requirements should be understood by the clients as well as the developers.

Modifiable

- A requirements document is modifiable if its structure and style are such that changes can be made easily, completely and consistently:
 - Easy to use organization – table of contents, index and cross references.
 - No redundancy – a requirement should not appear in more than one place.

Traceable

- Each requirement should be contained in a single, numbered paragraph so that it may be referred to in other documents:
 - Backward traceability - implies that we know why every requirement exists
 - Each requirement explicitly references its source in previous documents.
 - Forward traceability – all documents to follow will be able to reference each requirement.

Traceable (cont.)

- Example:
 - 2.1 Functional requirements:
 - 2.1.1 TVRS initialization:
 - ...
 - 2.1.15 TVRS shall display the user login window (see section 2.1.2.2)
 - 2.1.2 TVRS user interfaces:
 - 2.1.2.1 All user interaction with the TVRS shall occur by means of a graphical user interface.
 - 2.1.2.2 User login window:
 - » ...

Agenda

- Definitions
- Characteristics of requirements.
- Types of requirements
- Examples:
 - Traffic Violation Reports System

Types of requirements

- Functional
 - User interface structure and behavior
 - Inputs
 - Output
 - Data processing
 - Accuracy, precision...
 - Error handling
 - Initialization / Shutdown
- Non-Functional
 - Physical environment
 - Security (users...)
 - Performance
 - Cost
 - Portability
 - Reusability
 - ...and many more

Types of requirements (cont.)

- Functional requirements
 - Describe fundamental functions of the system.
 - System services which are expected by the user of the system.
- Non functional requirements
 - Constraints on the system.
 - Reliability, portability, safety, performance...
 - Application domain: interface with existing systems in the organization.
 - Two products could have exactly the same functions but their attributes can make them entirely different products.

Types of requirements (cont.)

- Measurable Non-functional Requirements
 - How can you test whether a system is “fast”, “secured”, or “maintainable” ?
 - Every product must have these attributes.
 - The degree to which each attribute is required should be documented:
 - Performance – response time, throughput, capacity
 - Efficiency – maximum load on resources.
 - Reliability – mean-time to failure (MTTF), full recovery time.
 - Usability – reports per hour, training.

Requirements Pitfalls

- Don't add unnecessary requirements
 - Writing requirements is a difficult task.
 - Increases complexity.
 - Invest the time in improving / validating the necessary requirements.
- Never assume the existence of external systems.
 - Must be explicitly required by the client.
- The main goal is to increase profits...
 - Avoid complex and/or expensive solutions

Agenda

- Definitions
- Characteristics of requirements.
- Types of requirements
- Examples:
 - Traffic Violation Reports System

Example: TVRS Requirements

- 1. Functional Requirements
 - 1.1. System initialization:
 - 1.1.1. TVRS shall automatically connect with the policemen, vehicles and offenders data bases according to the TVRS configuration file located at the root directory of the TVRS application.

Example: TVRS Requirements

- 1. Functional Requirements
 - 1.1. System initialization:
 - ~~1.1.1. TVRS shall automatically connect with the policemen, vehicles and offenders data bases according to the TVRS configuration file located at the root directory of the TVRS application.~~
 - 1.1. System initialization:
 - 1.1.1. TVRS *shall* automatically connect with the policemen, vehicles and offenders data bases.

Example: TVRS Requirements

– 1.2. User Interface:

- 1.2.1. All user interaction with TVRS shall take place by means of a graphical user interface.
- 1.2.2. User login window
 - 1.2.2.1. The user login window allows TVRS *operators* and *administrators* to log into the system in a secured fashion.
 - 1.2.2.2. The user login window shall allow the *user* to enter his login name and password.

Example: TVRS Requirements

- 1.2.2.3. The user login window shall provide users with the ability to choose between the following options:
 - » Entering the TVRS system (initiating user login)
 - » Shutting down the TVRS system (shutdown).
- 1.2.2.4. TVRS shall allow user to login if and only if all of the following holds (in the specified order)
 - » User entered his login name and password.
 - » User requested to enter the system.
 - » The login name is stored in TVRS and the password matches the password associated with that login name in TVRS.
- 1.2.2.5. In case all the conditions stated in section 1.2.2.4 hold, TVRS shall display the Main Menu window and allow the user access to it.

Example: TVRS Requirements

- 1.2.2.6. In the following cases, user login shall be *aborted* by TVRS:
 - » User login name or password are invalid (see section ...)
 - » User login name is not stored in TVRS.
 - » User password does not match the one associated with the given user name.
- 1.2.2.7. In case of login abortion, TVRS shall perform the following (in the specified order):
 - » Delay for a period of no less than 5 seconds and no more than 10 seconds. In this period of time, no interaction will be carried out with the user.
 - » Display an error message stating the reason of login failure.
 - » Enable interaction with user as specified in section 1.2.2.3

Example: TVRS Requirements

- 1.3. System Inputs:
 - 1.3.1. Traffic Violation details:
 - 1.3.1.1. A traffic violation shall include the following details:
 - » Violation id
 - Consists of decimal digits only.
 - Unique among all other traffic violation IDs.
 - » The Id of the policeman who issued the report (see ...).
 - » ...
 - 1.3.1.2. All details but the violation's description are mandatory.

Example: TVRS Requirements

- 1.4. System Outputs:
 - 1.4.1. Traffic Violations Report:
 - 1.4.1.1. Traffic violations shall be displayed in a table.
 - 1.4.1.2. Each traffic violation shall occupy a single row in the table.
 - 1.4.1.3. The following details shall be displayed for each traffic violation:
 - » Violation id.
 - » ...

Example: TVRS Requirements

- 1.4.1.4. Violations may be sorted according to the following criteria:
 - » Violation Id – ascending or descending order.
 - » Date – ascending or descending order (default criteria)
- 1.4.1.5 List of displayed violations may be filtered as follows:
 - » All violations
 - » Violations given by a specific policeman.
 - » Violations given in a period of time (single day resolution)
 - » ...

Example: TVRS Requirements

- 2. Non-functional requirements
 - 2.1. Reliability
 - 2.1.1. TVRS shall store all data in 2 different locations at distance of no less than 50 KM between them.
 - 2.1.2. TVRS will back-up all data automatically at 24:00 every night.
 - ...
 - 2.2. Security
 - 2.2.1. All data communication to/from the TVRS system shall be carried out over the secured private police network.
 - ...