Software Requirements
The Volere Requirements

Source:

"Mastering the Requirements Process"

S. Robertson, J. Robertson, 3rd Edition

Based on Eshcar Hilel

Agenda

- Definitions
- Quality Gateway
- The Volere template
- The Volere template example:
  - *Traffic Violation Reports System*

Definitions

- A **requirement** is something the system is capable of doing or a property that the system must have.
- The **requirements** must be discovered and specified before starting to build the product.

Definitions

- The **client** pays for the development of the product. (Computer center - traffic department)
- The **customer** buys the product once it is developed. (Sometimes the same as…)
- **Users** will ultimately operate the product. (Police officer)
- **Stakeholders** are people who have an interest in the product (system developers, QA people, lawyers etc.)

Product Purpose

- The product **purpose** describes the reason for building the product, it is the highest level customer requirement
- The **goal** (**purpose**) of the product, not only solves the problem, but also provides some advantage. PAM – Purpose, Advantage, Measurement
  - *Purpose: To allow on-line update of traffic violation*
  - *Advantage: To reduce road accident by immediately punishing "heavy" offenders*
  - *Measurement: Accidents attributed to human factor shall be no more than 15% of the total number of accidents*

Functional requirements

- Specify what the system must do
- Actions the product must take
- Derived from main goal of the product
- Not a quality
- Characterized by verbs
  - Example: *TVRS shall automatically connect with the policemen, vehicles and offenders data bases*

Non-functional requirements

- Properties, or qualities, that the system must have
- Characterized by adjectives
- Checklist:Look and feel, Usability, Performance Maintainability and Portability, etc
  - Example: *The interface between the user and the TVRS must have a maximum response time of two seconds*

Constraints

- Global issues that shape the requirements.
- They refer to any limitations on the way the product is produced
- Design solution that must be used, availability of time and money for the solution
  - Example: *TVRS must be a hand-held device*

Fit criteria

- Makes requirements measurable thereby testable
  - *Description: TVRS shall register traffic violations*
  - *Fit criteria: the registered traffic violations shall match information sent by the police officer*
  - *Description:TVRS shall be intuitive and self explanatory*
  - *Fit criteria: a police officer shall be able to use the product within 10 minutes of encountering it*
- The specification must contain a definition of the terms used in the fit criteria

Quality Gateway

Quality Gateway

Examine each requirement before entering the specification:

- Completeness
- Traceability
- Consistency
- Relevancy
- Correctness
- Ambiguity
- Viability
- Deal only with the problem
- Gold Plating

Completeness

- 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).

Traceability

- 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

Consistency

- 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.
  - Is every reference to a defined term consistent with its definition:
    - In one part: “*A policeman ID shall consist of decimal digits only*”, while in another part “*in case the policeman ID consists of non-alphanumerical characters, display an error message*”.

Consistency(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?

Relevancy

- Does this requirement contribute to the purpose of the product?
- Is every requirement relevant within the system boundaries?
  - *TVRS shall record the overtime worked by the police officers*

Correctness

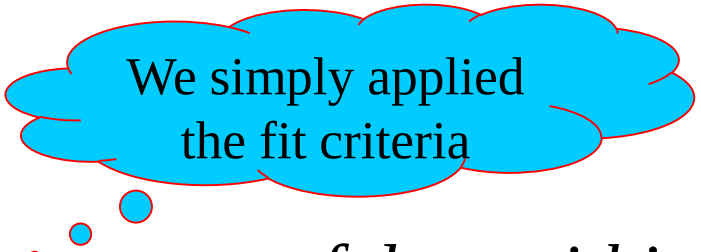
- Each requirement statement accurately represents the functionality required of the system to be built.
- Example (of an incorrect requirement):
  - Problem domain (real life) states that policeman ID numbers are in the range [10000…) and the requirements document specifies that each policeman has an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

Ambiguity

- 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”).
- The fit criteria is a quantification of the requirement, which can be used to test the solution.

Ambiguity(cont.)

- Examples (of ambiguity):
  - *The TVRS shall complete storage of data within a reasonable time of the user confirming a "Save" sequence.*
- Disambiguation:
  - *The TVRS shall complete storage of data within 5 seconds of the user confirming a "Save" sequence, 80% of the time.*



Viability

- Viable requirements are those that comply with the project's constraints.
- Do you have the technological skills to build the requirement?
- Do you have the time and the money to build the requirement?
- Is the requirement acceptable to all stakeholders?

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, for example: constraints part.
- The more abstract the requirement, the less likely it is to be a solution.
- Requirements should be understood by the clients as well as the developers.

Requirement or Solution?

- Solution: *The product shall have a clock on the menu bar.*

- Requirement: *The product shall make the user aware of the current time.*

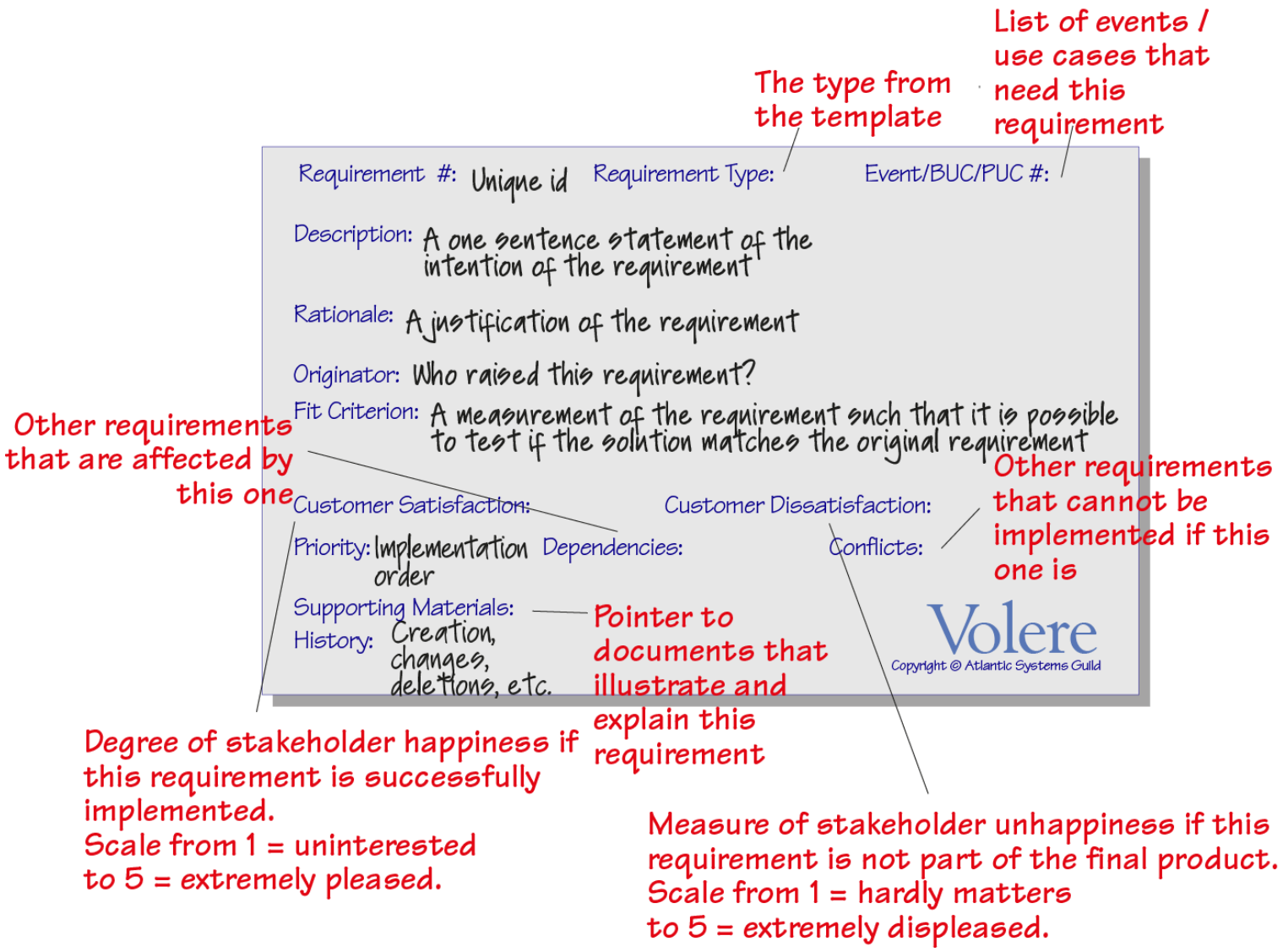
- Solution : *Users shall use passwords to access the system.*

- Requirement: *The product shall provide access to confidential information only to authorized users.*

Gold Plating

- The term comes from gold plated bathroom taps.
  - Example: *TVRS will play a piece of classical music during initialization*
- Does it matter if this requirement is not included?
- Sometimes a little gold plating makes a big difference to the acceptance of the product

Snowcard


List of events / use cases that need this requirement
The type from the template
Requirement #: Unique id
Requirement Type:
Event/BUC/PUC #:
Description: A one sentence statement of the intention of the requirement
Rationale: A justification of the requirement
Originator: Who raised this requirement?
Fit Criterion: A measurement of the requirement such that it is possible to test if the solution matches the original requirement
Other requirements that are affected by this one
Customer Satisfaction:
Customer Dissatisfaction:
Other requirements that cannot be implemented if this one is
Priority:
Implementation order
Dependencies:
Conflicts:
Supporting Materials:
History: Creation, changes, deletions, etc.
Pointer to documents that illustrate and explain this requirement
Volere
Copyright © Atlantic Systems Guild
Degree of stakeholder happiness if this requirement is successfully implemented.
Scale from 1 = uninterested to 5 = extremely pleased.
Measure of stakeholder unhappiness if this requirement is not part of the final product.
Scale from 1 = hardly matters to 5 = extremely displeased.

Requirements specifications

Outline of the “Volere Template”

Volere Template – part 1

- 1- The Purpose of the Project
- 2- The Stakeholders
- 3- Mandated Constraints

Volere Template – part 2

- 4- Naming Conventions and Terminology
- 5- Relevant Facts and Assumptions
- 6- The Scope of the Work
- 7- Business Data Model and Data Dictionary

Volere Template – part 3

- 8- The Scope of the Product
- 9- Functional Requirements
- 10- Look and Feel Requirements
- 11- Usability and Humanity Requirements
- 12- Performance Requirements

Volere Template – part 4

- 13- Operational and Environmental Requirements
- 14- Maintainability and Support Requirements
- 15- Security Requirements
- 16- Cultural Requirements
- 17- Compliance Requirements
- 18- Open Issues

Volere Template – part 5

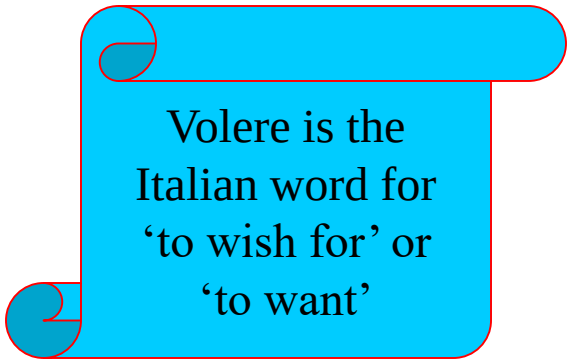
- 19- Off-the-Shelf Solutions
- 20- New Problems
- 21- Tasks
- 22- Migration to the New Product
- 23- Risks
- 24- Costs

Volere Template – part 6

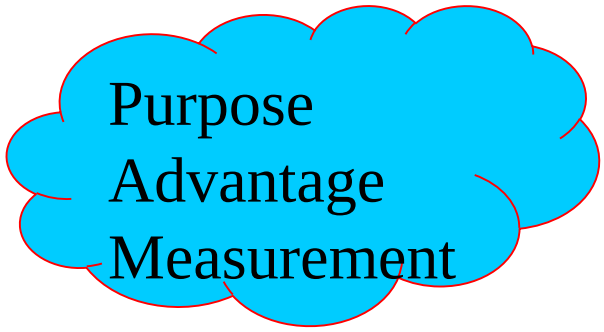
- 25- User Documentation and Training
- 26- Waiting Room
- 27- Ideas for Solutions

The Volere Requirements Specification Template

Case study:
TVRS Requirements



TVRS Requirements



Product Constraints

1. The purpose of the product
   1.1 The user problem or project background
   1.2 Goals of the product
2. Client, Customer and Stakeholders
3. Users of the product

TVRS Requirements

4. Requirements Constraints
   - 4.1 Solution constraints
     - 4.1.1 *TVRS must be a hand-held device*
     - 4.1.2 *TVRS must use the Windows NT operating system*
   - 4.2 Implementation environment
   - 4.3 Partner applications
   - 4.4 Commercial off-the-shelf software
   - 4.5 Anticipated workplace environment
     - 4.5.1 *The workplace of the Police officer is outside*
   - 4.6 Project schedule
   - 4.7 Project budget

TVRS Requirements

5. Naming Conventions and Definitions
6. Relevant Facts
7. Assumptions

TVRS Requirements

Functional Requirements

8. The scope of the product
   - 8.1 The context of the work
   - 8.2 Work partitioning
   - 8.3 Product Boundary

TVRS Requirements

9. Functional and Data Requirements

9.1. *System initialization*:

9.1.1. *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.*

9.1. System initialization

9.1.1. *TVRS shall automatically connect with the policemen, vehicles and offenders data bases.*

TVRS Requirements

9.2. Functionality

9.2.1. *TVRS enables on-line recording of traffic violations*

9.2.2. *The recorded traffic violations shall match information sent by the police officer*

9.2.3. *TVRS enables on-line access to previous traffic violations information of a given driver*

...

TVRS Requirements

9.3. System Inputs

9.3.1. *Traffic Violation shall include the following details*:

9.3.1.1. *Violation id:*

Consists of decimal digits only.

Unique among all other traffic violation IDs.

9.3.1.2. *The Id of the policeman who issued the report*

…

9.3.1.12. *All details but the violation's description are mandatory*

TVRS Requirements

9.4. System Outputs:

9.4.1. *Traffic Violations Report*:

9.4.1.1. *Traffic violations shall be displayed in a table.*

9.4.1.2. *Each traffic violation shall occupy a single row in the table.*

9.4.1.3. *The following details shall be displayed for each traffic violation:*

Violation id.

…

TVRS Requirements

Non-functional requirements

10. Look and Feel Requirements

10.1 *The product shall appear authoritative*

10.2 *The product shall use the company colors and fonts*

11. Usability Requirements

11.1 Ease of use

~~11.1.1 *TVRS shall use a mouse*~~

11.1.1 *TVRS shall allow the user to directly manipulate all interface items.*

11.2 Ease of learning

11.2.1 *A police officer shall be able to add TV within 10 minutes of encountering the product*

TVRS Requirements

12. Performance Requirements

12.1 Speed requirements

12.1.1 *The interface between the user and the TVRS must have a maximum response time of two seconds*

12.2 Safety critical requirements

12.3 Precision requirements

12.4 Reliability and availability requirements

12.4.2. *TVRS will back-up all data automatically at 24:00 every night*.

12.5 Capacity requirements

TVRS Requirements

13. Operational Requirements

13.1 Expected physical environment

13.2 Expected technological environment

13.3 Partner application

13.3.1 *TVRS must interface with the application that run on the remote police station*

TVRS Requirements

14. Maintainability and Portability Requirements

14.1 Easy maintenance quantification

14.2 Special conditions for maintenance

14.2.1 *The maintenance releases will be offered to end-users once a year*

14.3 Portability requirements

14.3.1 *TVRS is expected to run under Windows NT and UNIX*

TVRS Requirements

15. Security Requirements

15.1 Is the product confidential

~~15.1.1 *TVRS shall ask for a password when data is accessed*~~

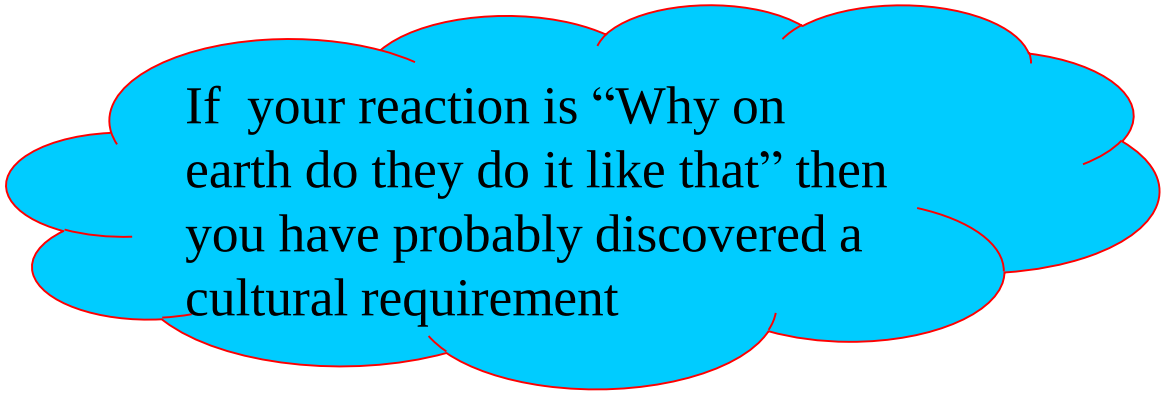
15.1.1 *TVRS shall ensure that the data can be accessed only by authorized users*

15.2 File integrity requirements

15.2.1. *All data communication to/from the TVRS system shall be carried out over the secured private police network.*

TVRS Requirements

16. Cultural and Political Requirements
17. Legal Requirements



TVRS Requirements

Project Issues

18. Open Issues
19. Off-the-Shelf solution
    - 19.1 Ready made product to buy
    - 19.2 Ready made components to use
    - 19.3 Something to copy
20. New problems

TVRS Requirements

21. Tasks
    21.1 Steps to deliver the product
    21.2 Development phases
22. Cutover
    22.1 Get existing procedures to work for the new product
    22.2 Modify/translate data for the new product

TVRS Requirements

23. Risks
    - 23.1 Risks you face
    - 23.2 Contingency plans
24. Costs
25. User Documentation Plan
26. Waiting Room
    - 26.1 *Input will be received through a VUI (Voice User Interface)*

Links

- **https://www.volere.org/**
- **Mastering the requirements process**
  - **https://www.youtube.com/watch?v=6Sdpa6Hp40A**
- **Mastering the Requirements Process: Volere Flow**
  - https://www.youtube.com/watch?v=WywPA6npWBU