Software Testing

Static Testing – Reviewing Requirements

(Are you SMART?)

by J. Janvier (Jay/文宇)

Reviewing Requirements

Reviewing of Requirements

Requirements must be SMART:

S = Specific

M = Measurable

A = Achievable

R = Relevant

T = Traceable

"Maybe I should've been more clear when I said I wanted to build a bridge between business and IT."

Reviewing Requirements

Specific:

- Not generic, (watch for words like: soon, fast, later)
- Not open to miss-interpretation when read by others, (‘open’ system)
- State only 1 requirement in the statement. (watch for words like: and, or, but)

The report should display all the monthly data from the marketing department

Good or bad requirement? Why? **Bad requirement: What is ‘all data’?**

Good requirement:
The report shall contain the following columns: Total Sales for Month, Average Retail Price, Total Units Sold, Remaining Inventory, Total Cost of Goods Sold.

Reviewing Requirements

Measurable:

- It must be possible to verify the completion,
- What is the acceptance criteria?

(watch for words like: best, optimal, fastest, around, about)

The system shall have an optimal response time for the end-user.

Good or bad requirement? Why?

Bad requirement: What is ‘optimal’?, Can it be measured?

Good requirement:
The system shall have user response times on user click-events that are 5-seconds or less during business hours of 9AM-5PM, Mountain Time, Monday-Friday.

Reviewing Requirements

Achievable:

- requirement must physically able to be achieved given existing circumstances.
- Must be realistic

The resulting web site shall be so popular that it gets 1,000,000 hits within the first week.

Good or bad requirement? Why?

Bad requirement: It is possibly a wish from the customer, but both parties know it is not achievable.

Good requirement:
The resulting web site, when reviewed by XYZ site-rating agency, will receive at least a 3 out 5 five rating in the category of “Fun websites to visit.”

Reviewing Requirements

Relevant:

- Is the requirement a requirement?
- Duplicate requirements

The Technical Support Team of the Customer will install the application on the servers.

Good or bad requirement? Why?

Bad requirement: It is a statement/ part of a deployment plan. It is not a functional or non-functional requirement to the software

Reviewing Requirements

Traceable:

- All requirements must have a unique ID.
- A requirement must always be traced back to a customer requirement (must include a reference to ID('s) of customer requirement(s).
- It prevents scope creep.
- It prevents bad surprises at the hand-over/CAT.

(Customer Level Requirement)

CR-UI-342: The system shall use the company logo and house-style colours.

(Lower level Requirement)

The system shall use blue (RGB 234,132,45) backgrounds in the screens in the user interface. (Reference CR-UI-342)

Good or bad requirements? Why?

Bad requirements: The second requirement does not have an ID. The first requirement contains multiple requirements in one statement. The first requirement does also not specify what the house-style colours are, so not measureable.

Reviewing Requirements

Separation of Functional and Non-Functional Requirements:

Functional and Non-Functional Requirements are, in most cases, differently (discussed later in the course)

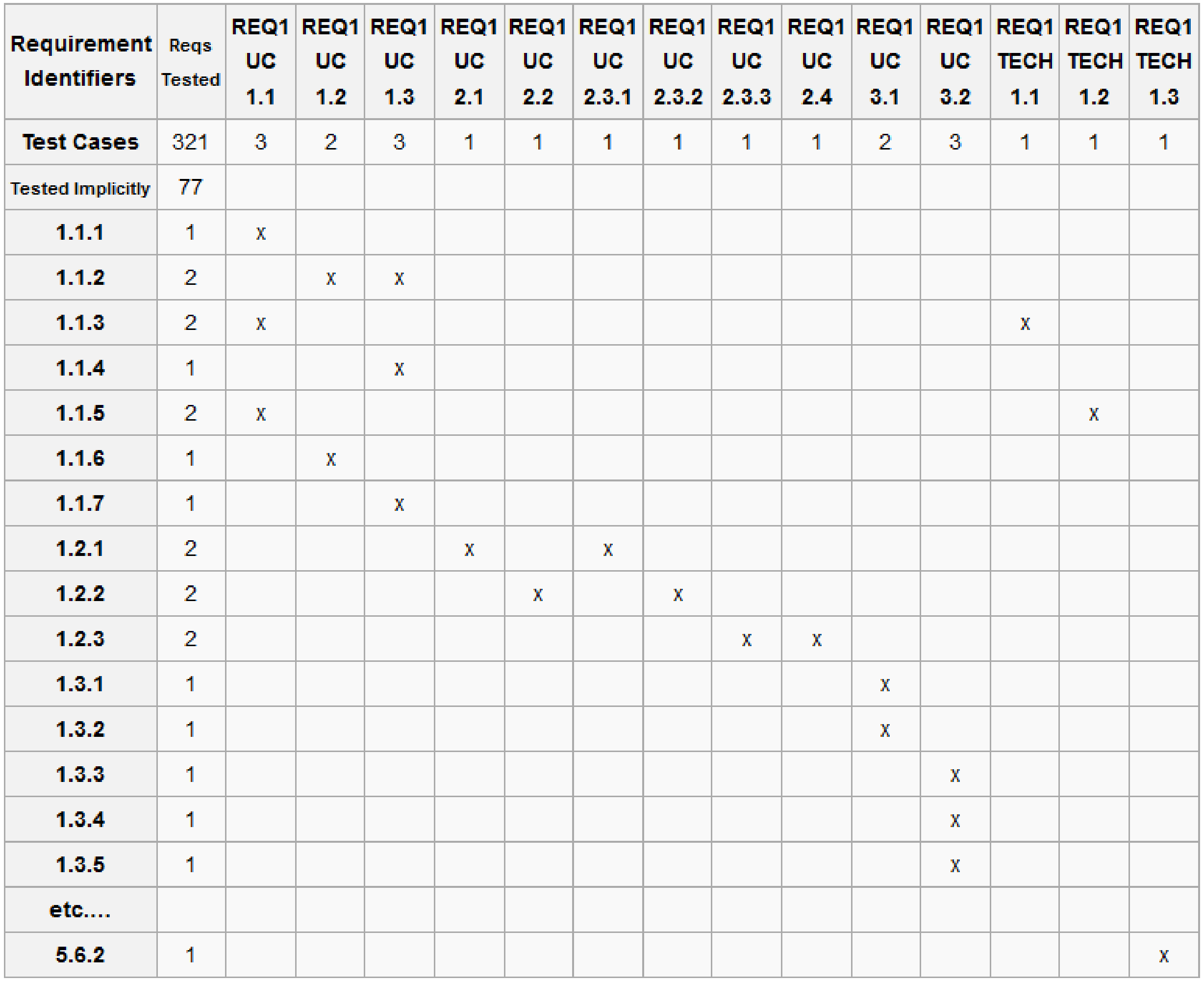
Requirements must have ID’s:

Most projects will end up with a levelled hierarchy of requirements, with increasing detail: Customer Requirements -> System Requirements -> Subsystem Requirements
Every single lower level requirement should be referring to a higher level requirement.
A structured system of ID’s would help a project with this traceability.
Tools, like Telelogic Doors, also provide this functionality by default.

Traceability Matrix:

Matrices which give an overview of the links between requirements, will help tester to determine, whether all requirements can be traced.
Testers might also create traceability matrices between Tests and Requirements.

Traceability Matrix Example



Requirement Identifiers	Reqs Tested	REQ1 UC 1.1	REQ1 UC 1.2	REQ1 UC 1.3	REQ1 UC 2.1	REQ1 UC 2.2	REQ1 UC 2.3.1	REQ1 UC 2.3.2	REQ1 UC 2.3.3	REQ1 UC 2.4	REQ1 UC 3.1	REQ1 UC 3.2	REQ1 TECH 1.1	REQ1 TECH 1.2	REQ1 TECH 1.3
Test Cases	321	3	2	3	1	1	1	1	1	1	2	3	1	1	1
Tested Implicitly	77														
1.1.1	1	x													
1.1.2	2		x	x											
1.1.3	2	x											x		
1.1.4	1			x											
1.1.5	2	x												x	
1.1.6	1		x												
1.1.7	1			x											
1.2.1	2				x		x								
1.2.2	2					x		x							
1.2.3	2								x	x					
1.3.1	1										x				
1.3.2	1										x				
1.3.3	1											x			
1.3.4	1											x			
1.3.5	1											x			
etc....															
5.6.2	1														x

Questions?