

Atomic Requirements: where the rubber hits the road



This is the sixth article in a series that explains the thinking behind the *Volere*¹ requirements techniques. Subsequent articles will explore various aspects of applying these techniques in your environment.

*By Suzanne Robertson & James Robertson
The Atlantic Systems Guild – December 2009*

What are Atomic Requirements?

When you have a requirement that is measurable, testable, traceable and detailed enough to define all aspects of a need without further breakdown then you have an atomic requirement. You can think of the atomic requirements as the lowest level requirements. In fact the collection of atomic requirements specifies what we require the solution to do.

When there are too many atomic requirements to hold simultaneously in a human brain – say around 10 on a good day - then there is a need for higher-level groupings of requirements. The higher-level groupings of requirements that you choose to maintain depend on the communication points that you have in your project. Some useful high-level groupings are Business Use Cases (BUCs), Product Use Cases (PUCs), Work Scope, Product Scope, Features, Components, Sub Components. Whichever high level groupings you use it must be possible to trace each atomic requirement to the higher-level groups that include it.

¹ *Volere* is the Italian verb – to wish or to want

Why have Multiple Attributes?

An atomic requirement concerns a number of different stakeholders for a number of different reasons. In order to understand things like where it comes from, what it means, who cares about it, why it matters, how to test it, what else will be affected by changes the requirement needs a number of attributes. The business person or subject matter expert focuses on a business description of the requirement. The developer wants to know why the requirement is important along with a precise quantification of what it means. The tester needs to have a testable measurement of the requirement. The business analyst needs to know how each requirement affects other requirements. The maintainer needs to know how this requirement affects requirements that are already implemented.

Each stakeholder has a stake in particular attributes of the requirement according to his role and responsibility.

The Attributes

The Volere Requirements Shell, also referred to as a Snow Card, identifies a number of attributes that together compose one atomic requirement.


Requirement #:	Requirement Type:	Event/Use Case #'s:
Description:		
Rationale:		
Originator:		
Fit Criterion:		
Customer Satisfaction:	Customer Dissatisfaction:	
Priority:	Conflicts:	
Supporting Materials:		
History:		
 Copyright © Atlantic Systems Guild		

Figure 1: The Volere requirements shell identifies all the attributes of an atomic requirement.

The following provides some notes on each of the attributes of an atomic requirement.

Requirement Number

A unique identifier for this requirement. The *only* purpose of this identifier is as an identifier for *this* requirement. Do not include hybrid meanings (like the status, or the importance or whatever) in this number – it will only end in tears. If you want to record other facts about the requirement then have a separate attribute.

Requirement Type

The type of the requirement as defined in the Volere requirements template. In summary the types are: Functional, Constraint and Non Functional (there are many non functional sub types). The reason for requirements type is to help you find the requirements and to be able to group the requirements to facilitate the involvement of appropriate specialist stakeholders like usability experts, security experts, performance experts and so on.

Business Event Number

Traces this requirement back to the relevant business event/s that define the scope of the work that is relevant to this project.

Business Use Case (BUC)

Traces this requirement to the business use case/s that contain this atomic requirement.

Product Use Case (PUC)

Traces this requirement to the product use case/s to which this requirement has been allocated.

Description

A one sentence description that states this requirement from the perspective of someone who is concerned with this part of the work/business.

Rationale

A justification of why this requirement is important. This should help to identify the expected benefit of having a solution to this requirement.

Originator

The person who raised this requirement.

Fit Criterion

A measurement of the requirement such that it is possible to non-subjectively test whether the solution fits the original requirement.

Customer Satisfaction

On a scale from 1 to 5 how much benefit will result if a solution that fits this requirement is delivered?

Customer Dissatisfaction

On a scale from 1 to 5 how much damage will there be if we cannot deliver a solution that fits this requirement?

Customer Satisfaction and Customer Dissatisfaction are a tool for helping people to think about the value of the requirement and keeping them aware that the definition of a requirement does not necessarily guarantee a solution.

Priority

The decision on what priority is assigned to the requirement. This is quantified using the scale that best suits your organisation. For example you could use: High, Medium, Low OR 1,2,3, OR (MoSCow) Must, Should, Could, Won't OR Implementation Version Number OR any other scheme that suits you.

Conflicts

Other requirements that cannot be implemented if this requirement is implemented as specified. The purpose of this is to encourage people to identify decisions and choices that need to be made before implementation.

Supporting Materials

Pointers to documents, examples and people that illustrate and can explain more about the background of this requirement.

History

The attributes that you choose to keep track of to trace the lifecycle of this requirement within your organisation. For example you might have checkpoints for requirements like: project phases, review checkpoints, inclusion in documents, name of approver, date of approval, date of implementation and so on. The number of historical attributes you choose to

record varies depending on the size, complexity and fragmentation of your project.

Dependencies

We keep track of the dependencies between requirements by maintaining the discipline of a data dictionary. Every term used in a requirement is defined in the dictionary and we check that the terms are being used consistently within the project. The dictionary acts as a central communication point for the project team.

Atomic Requirement Example

The following is an example of an atomic requirement for the Library Loans case study that we have used in other articles in this series.

Requirement #:	29	Requirement Type:	Functional	Event/Use Case #'s:	1/1..3
Description:	The product shall identify all outstanding books on loan to a Borrower				
Rationale:	Need to know which book loans are overdue when making a decision about whether to extend a loan.				
Originator:	Aislinn Weatherspoon, Chief Librarian				
Fit Criterion:	The Outstanding Book Loans for a Borrower are those where the Loan Expiry Date is before or equal to Today's Date				
Customer Satisfaction:	5	Customer Dissatisfaction:	5		
Priority:	VI	Conflicts:			
Supporting Materials:	Library loan conventions paper January 2008				
History:	June 25, 2009 Passed Quality Gateway review				

Volere
Copyright © Atlantic Systems Guild

Figure 2: An example of the attributes of an atomic requirement.

The example in figure 2 shows the attributes for a functional requirement. Note how each of the attributes adds to the understanding of the details of the requirement. The customer satisfaction and dissatisfaction are both 5 which indicates that this is a core requirement. In other words if it is not met then there is no point in building the product.

Last Word

People often ask whether we recommend writing requirements onto physical snow cards. If you are running a requirements discovery workshop then the actual cards are a useful device. They also work rather well if you want to have a detailed discussion about a specific requirement. However, given the volume of requirements and the complexity of keeping track of levels it makes sense to use automation for completeness and consistency checking and for maintaining traceability. There are many requirements tools on the market, many are summarised at <http://www.volere.co.uk/tools.htm>

Many business analysts use a combination of a word processor, a spreadsheet and a drawing tool to automate parts of their requirements specification. Other people make use of their existing database software or their Intranet. If you have a consistent understanding of the attributes of atomic requirements and their traceable connections to higher level groupings then you are in a position to choose what to automate given the facilities available to you.

More information is available:

- <http://www.volere.co.uk>
- in three books written by James Robertson & Suzanne Robertson, the most relevant to this article is *Mastering the Requirements Process – second edition*.
- in Volere seminars and consulting
- on the Volere Requirements Linked In group
<http://www.linkedin.com/e/vgh/2491512/>

Previous articles are available at <http://www.volere.co.uk>

Suzanne Robertson and James Robertson are principals and founders of The Atlantic Systems Guild <http://www.systemsguild.com> and joint originators of the **Volere** requirements process, template, checklists and techniques <http://www.volere.co.uk>

You can contact them at
suzanne@systemsguild.net
james@systemsguild.net