The Atlantic Systems Guild Ltd.

Agile Requirements

This is number eight in a series that explains the thinking behind the *Volere*[1] requirements techniques. Previous and future articles explore various aspects of applying these techniques in your environment.

By Suzanne Robertson & James Robertson
The Atlantic Systems Guild — August 2010

Origin of Agility

If you work in systems development, especially if you are developing software, you have almost certainly been exposed to the idea of being Agile. However, possibly you are not aware of from the origin of this idea. The agile manifesto is a set of principles, and is the result of a collaboration between a group of systems engineering specialists that started in 2001. The manifesto signatories realised that some of our more heavy-handed software development methodologies and documentation were slowing us down. They reasoned that many aspects of the way we were developing software were preventing us from carrying out the purpose of our real jobs: to deliver valuable software that helps our customers to do their work.

Since the publication of the manifesto, systems developers in organisations all over the world have committed to applying the agile principles to the way that they build systems. More recently, business analysts and requirements specialists have been considering how the agile principles can be applied when discovering and communicating business requirements.

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

Users of the Volere requirements techniques often discuss how the techniques can help them to be more agile. That is, to deliver valuable and relevant software more quickly in their environment? During a recent discussion about this subject on the Volere LinkedIn group http://www.linkedin.com/e/vgh/2491512/ John Palmer suggested that it would be valuable to draw parallels between the agile principles and how Volere techniques address each one of them. This article will do that.

Addressing the Agile Principles

The following is a list of each of the *Agile Principles (in italics)*, followed by the ways that Volere techniques address that principle.

Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
Volere is based on well-established systems engineering principles. The Knowledge Model and Template provide a linguistic structure for defining and connecting your requirements knowledge at a number of levels. Having a well-defined structure to the requirements gives the business analysts and developers the freedom to work in whatever order or to whatever level of requirements detail will facilitate the earliest delivery of the correct software.

Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
It is difficult, if not impossible, to deal with changing requirements unless you know why you are changing and what you are changing from. The Volere techniques provide quick and communicable ways of keeping track of what you know and being able to recognize and integrate changes regardless of when they happen.

Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
The Knowledge Model identifies large and small functional chunks. These can be easily identified and can be delivered independently and iteratively, At the same time, the developers can keep track of how they connect to other parts of the business problem.

Business people and developers must work together daily throughout the project.

The important part of this principle is communication. The way that Volere requirements are composed means that both parties can understand the requirements. At the same time, the idea of the Fit Criterion means that both parties arrive at an identical understanding of the requirement. Similarly the Volere viewpoints give all parties a common language to express their ideas for the system.

Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
Project sociology techniques support the differences between individuals and the importance of each project group making a strategy that suits its own sociological mix.

The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
Volere is content driven, not form driven. The Volere approach is, whenever possible, to talk face-to-face. However, conversations have a way of being forgotten or misremembered. The Volere approach provides various ways of recording the essential information from the conversations without placing a documentation burden on the participants. The use of scenarios, low-fidelity prototypes and the like mean that in many cases, the output from the conversation is able to be sufficient specification for the correct development.

Working software is the primary measure of progress.
Of course. Volere makes all requirements testable so that "working" can be proved. This is also helped by the integration of prototyping and simulation at any stage of your development process.

Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
Volere supports this by encouraging an iterative development cycle that includes all participants. Business analysts work with the relevant stakeholders on small slices of the business problem allowing software development to proceed at the earliest opportunity. By thoughtful partitioning of the business problem, the team are able to establish a sustainable delivery rhythm.

Continuous attention to technical excellence and good design enhances agility.

Volere techniques provide ways of articulating design alternatives, comparing and choosing the best and recognizing good design patterns.

Simplicity--the art of maximizing the amount of work not done--is essential.
Volere techniques focus on identifying the essence of the problem and prioritising to provide the highest business value.

The best architectures, requirements, and designs emerge from self-organizing teams.
Volere provides the means for the self-organizing team to communicate with the overall enterprise.

At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.
The Volere techniques address both sociological and technological concerns and prompt people to look for both technical and social reasons for success and failure.

Last Word

To maximise your potential for agility you need to be able to distinguish techniques that support your ability to deliver valuable software and those that inhibit you. Any technique that forces you to do things in a prescribed order, to an unchanging level of detail, in a specific form is a technique that will inhibit your agility.

The Volere techniques enhance your ability to be agile by supporting your thinking and communication with your colleagues and encouraging you to be innovative and extemporaneous. The techniques are based on solid systems engineering principles like: partitioning, seeing the problem from different points of view, communication of scope, linguistic elements appropriate to the domain, making abstractions, differing levels of detail. You can work in the way that is appropriate for your project and produce a knowledge trail as a result of doing your work.

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