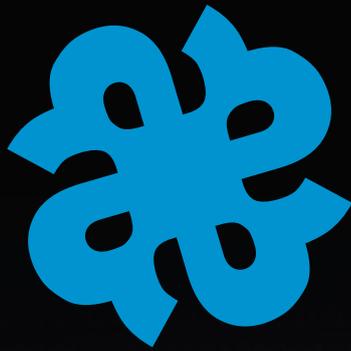


Text



# Agile Record

The Magazine for Agile Developers and Agile Testers



**Security Testing in an Agile Environment**

February 2014

[www.agilerecord.com](http://www.agilerecord.com)

free digital version

made in Germany

ISSN 2191-1320

**issue 17**

## Advanced Product Owners

by Tom & Kai Gilb

We are going to argue that the normally defined role of Product Owner (PO) is inadequate for projects that have serious multiple quality requirements, and consequent architecture processes, to deliver the necessary levels of performance and quality.

This includes all large serious projects, such as government or corporate projects. We do *not* want to argue that the Product Owner role as conventionally defined is inadequate for *small* projects, nor for projects that are *not* dependent on multiple state of the art quality, performance, and cost levels - and the consequent architecture to meet them. However, a point is reached where a project is so demanding that methods adequate for smaller projects will fail. The methods of a homeowner-built house will not work for a 200-story skyscraper. The Scrum project failure rate (about 19%) may be better than waterfall (more like 40%) [3] due to better feedback. But killing only one out of five pedestrians at a crossing, as opposed to two out of five, is still not good enough. We need to get closer to zero project failures in IT development. The myth is that the conventional PO process is *universal*, and that it scales up to any type of project. I believe we need to become more aware of the limits of today's

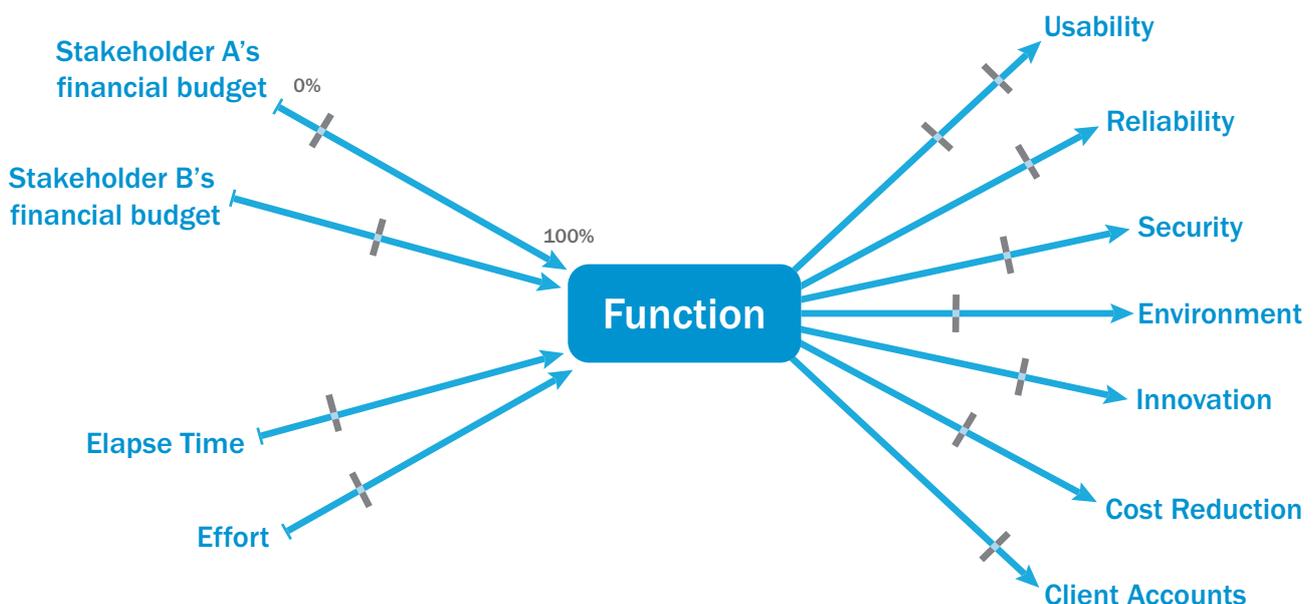
dogma, and to identify practices appropriate for Agile methods in the more demanding projects.

The myths of the process owner are in italics below (<http://www.mountaingoatsoftware.com/agile/scrum/product-owner>). Our comments will briefly indicate another point of view for the more demanding projects. This column will not allow us to argue in detail, but the references will help those who need more depth. The purpose of this column is to open up the debate for a less dogmatic and less oversimplified presentation.

### 1. *The Scrum product owner is typically a project's key stakeholder.*

[1] The PO in reality needs to perform at least three very distinct functions, which are well understood in larger scale software engineering, and enterprise IT.

**Requirements Engineering (RE):** this is NOT a matter of User Stories. This is primarily a matter of quantified specification of the top-level primary drivers of a project [5]: qualities such as security, usability, and adaptability [4]), project constraints, and performance requirements. There is no such concept as this in Scrum or other similar Agile variations at present.



The Advanced Product Owner function for complex projects needs to be able to manage multiple top level and critical performance improvement objectives, while managing multiple resources and constraints. Simple specifications such as User Stories and Burn Down Charts will not do this job.

**Architecture Engineering (AE):** this is the process of finding a set of high-level architectures or designs which will probably deliver (to be verified in each sprint by measurement of delivered value and of costs) the quality levels, and performance levels within resource constraints, as specified in the top-level requirements above [6]. There is no such concept as this in Scrum or other similar Agile variations at present.

**Dynamic Prioritization [7]:** deciding what to do next.

**2. Part of the product owner's responsibilities is to have a vision of what he or she wishes to build, and to convey that vision to the scrum team [1].**

Large projects have something like 40 or more stakeholders, each of whom has several requirements. No single individual can know of, and be responsible for, a complete vision. At best a single Requirements Engineer (or Business Analyst, or more likely a team) can collect the information from all stakeholders, and specify, validate, and refine it iteratively as the project progresses, and present it in a clear testable manner.

**3. This is key to successfully starting any Agile software development project [1].**

There are many 'keys' to starting, and successfully completing, an Agile project. The PO armed with User Stories is NOT a main key in the case of large-scale complex projects [2]. The main key (though there are several!) in our view is to measurably establish the top ten primary improvements that the project will deliver: the primary stakeholder values [5]. The User Stories are, in reality, detailed designs to deliver higher level values such as usability and security. The Scrum PO has no concept of this now. But Scrum Inc. has agreed that something like this is necessary. So it might happen.

**4. The Agile product owner does this in part through the product backlog, which is a prioritized features list for the product [1].**

Intelligent prioritization of large-scale and complex systems is a highly dynamic process which depends on very many factors [7]. The Business Analyst can at best collect a diverse set of information that can be used to determine priorities for the next sprint. But the numeric feedback from delivered sprints (what value was delivered, what it cost) needs to be applied to determine the current priority. We find that spreadsheet feedback from projects can be used to calculate current priorities. You do not need a PO, you need the key facts and measures of progress.

**5. The product owner is commonly a lead user of the system, or someone from marketing or product management, or anyone with a solid understanding of users, the market place, the competition, and of future trends for the domain or type of system being developed [1].**

For small projects we have no problem with this. For our large and complex projects we need, as stated above, professional

requirements competence to represent all critical stakeholder requirements, and to create professional architecture.

**6. The Scrum product owner's job is to motivate the team with a clear, elevating goal [1].**

Well, this is fine for simpler projects. But in larger projects there is always a set of key objectives, so the motivation needs to be to reach the set of clear objectives. We suspect that this job is best tackled by the Project Manager. But we agree that motivation is critical.

**7. Team members know best what they are capable of, and so they select which user stories from the top of the product backlog they can commit to delivering during any sprint [1].**

We agree that development team members are best placed to understand their delivery capability in a sprint. The PO need not be involved.

**8. The product owner role requires an individual with certain skills and traits, including availability, business savvy, and communication skills. First, the Scrum product owner needs to be available to his or her team. The best product owners show commitment by doing whatever is necessary to build the best product possible – and that means being actively engaged with their teams [1].**

For larger projects the concept of this single individual is under-dimensioned. The skills needed are the skills of multi-stakeholder requirements analysis, and then the skills of design architects and engineers to find appropriate technology to deliver the stakeholder values. This is a highly specialized and qualified team as discussed above. The current PO does not have the responsibility, the training, or the capability to do this well enough on larger projects, the ones that fail today.

**9. Business savvy is important for the agile product owner because he or she is the decision maker regarding what features the product will have. That means the Agile PO should understand the market, the customer, and the business in order to make sound decisions [1].**

This is properly the task of a marketing function, which is one of many stakeholders for larger projects.

**10. Finally, communication is a large part of the product owner's responsibilities. The product owner role requires working closely with key stakeholders throughout the organization and beyond, so he or she must be able to communicate different messages to different people about the project at any given time [1].**

Sounds good to us! But we need to specifically learn to communicate the many critical qualities which make up about 70% of stakeholder values. Current PO culture with current User Stories do not even recognize this task. It is too focused on churning out code and features. We need a culture that recognizes that code is not the center of the universe. It should be about building systems to deliver value to stakeholders. The current Product Owner and User-Story model is dangerously inadequate for that task. If people recognized that it has limits, and that we need an engineering culture to tackle the larger and more complex projects, we believe our profession could move towards the reasonable target of zero defects for project success. ■

## References

1. <http://www.mountaingoatsoftware.com/agile/scrum/product-owner>
2. User Stories: A Skeptical View, Agile Record, 2012, [www.gilb.com/dl46](http://www.gilb.com/dl46)
3. Jeff Sutherland's ABE Warsaw video, 2013, <https://www.youtube.com/watch?v=YUaywvsfe78>
4. Chapter 5: Scales of Measure: [http://www.gilb.com/tiki-download\\_file.php?fileId=26](http://www.gilb.com/tiki-download_file.php?fileId=26)
5. The Top 10 Critical Requirements are the Most Agile Way to Run Agile Projects, Agile Record August 2012. [http://www.agilerecord.com/agilerecord\\_11.pdf](http://www.agilerecord.com/agilerecord_11.pdf)
6. Agile Architecture Engineering: Dynamic Incremental Design Selection and Validation, Agile Record, Nov. 2013 Issue 13. [http://www.agilerecord.com/agilerecord\\_13.pdf](http://www.agilerecord.com/agilerecord_13.pdf)
7. Managing Priorities, <http://www.compaid.com/caiinternet/ezine/gilb-managingpriorities.pdf>

## > about the authors

### Tom Gilb and Kai Gilb



*Tom Gilb and Kai Gilb have, together with many professional friends and clients, personally developed the agile methods they teach. The methods have been developed over five decades of practice all over the world in both small companies and projects, as well as in the largest companies and projects. Their website [www.gilb.com/downloads](http://www.gilb.com/downloads)*

*offers free papers, slides, and cases about agile and other subjects. There are many organisations, and individuals, who use some or all of their methods. IBM and HP were two early corporate-wide adopters (1980, 1988). Recently (2012) over 15,000 engineers at Intel have voluntarily adopted the Planguage requirements specification methods; in addition to practicing to a lesser extent Evo, Spec QC and other Gilb methods. Many other multinationals are in various phases of adopting and practicing the Gilb methods. Many smaller companies also use the methods.*

### Tom Gilb

*Tom is the author of nine published books, and hundreds of papers on agile and related subjects. His latest book 'Competitive Engineering' (CE) is a detailed handbook on the standards for the 'Evo' (Evolutionary) Agile Method, and also for Agile Spec QC. The CE book also, uniquely in the agile community, defines an Agile Planning Language, called 'Planguage' for Quality Value Delivery Management. His 1988 book, Principles of Software Engineering Management (now in 20th Printing) is the publicly acknowledged source of inspiration from leaders in the agile community (Beck, Highsmith, and many more), regarding iterative and incremental development methods. Research (Larman, Southampton University) has determined that Tom was the earliest published source campaigning for agile methods (Evo) for IT and Software. His first 20-sprint agile (Evo) incremental value delivery project was done in 1960, in Oslo. Tom has guest lectured at universities all over UK, Europe, China, India, USA, Korea – and has been a keynote speaker at dozens of technical conferences internationally.*

*Twitter: [@mtomgilb](https://twitter.com/mtomgilb)*

### Kai Gilb

*Kai Gilb has partnered with Tom in developing these ideas, holding courses and practicing them with clients since 1992. He coaches managers and product owners, writes papers, develops the courses, and is writing his own book, 'Evo – Evolutionary Project Management & Product Development.' Tom & Kai work well as a team; they approach the art of teaching their common methods somewhat differently. Consequently the students benefit from two different styles.*