

The Agile Business Analyst: Eyes for Waste

By Ellen Gottesdiener

Copyright © EBG Consulting, Inc., 2009
EBG Consulting, Inc.: www.ebgconsulting.com

This article was first published in the February, 2009 Issue of *Modern Analyst*

As an agile analyst, you know how easy it is to get lost in a blizzard of project details. The way to keep focused is to remind yourself of the chief reason for using agile processes: to deliver business value for your organization. With that in mind, we'll look at specific ways you can combine two approaches: agile and lean.

Lean Focuses on Value

Lean processes—whether you're building bicycles, assembling TV dinners, or developing software—are all about value. *Value*, in lean terms, is any action or thought a customer is willing to pay for (Larman and Vodde 2009). Activities like rework, reprocessing, reformatting, storage, handling, and sign-offs are not valuable. In lean terminology, they're *waste*.

Shigeo Shingo (1981) identified seven wastes of manufacturing, and Mary and Tom Poppendieck (2007) translated these into waste in software development, as shown in Table 1.

Table 1: Translating the Seven Manufacturing Wastes into Software Development

The Seven Wastes of Manufacturing	The Seven Wastes of Software Development
Overproduction	Partially done work
Inventory	Extra features
Extra processing	Relearning
Transportation	Handovers
Motion	Task switching
Waiting	Delays
Defects	Defects

Source: Poppendieck & Poppendieck, 2007.

For an agile business analyst with keen eyes toward eliminating waste, this list offers a goldmine of opportunity.

Partially Done Work

Artifacts such as a detailed requirements specifications, untested or undocumented code, design drafts, or other work products not implemented—all are examples of partially done work. As an agile business analyst, what can you do?

- Help the product owner (a.k.a. customer) develop a product backlog with minimal detail.
- Aid the product owner and team to elaborate on backlog items only when they are to be pulled into the next iteration (a.k.a. sprint).
- Create “just enough” analysis artifacts so that your team can estimate and define the tasks for each iteration backlog item during the planning workshop.
- Anticipate when backlog items won’t be “ready” (not enough is known to plan them), and postpone work on them until it will produce value.

Extra Features

A feature is waste if you capture and implement it “just in case” (overspecification) or because the customer “might want it” (gold plating). It doesn’t add value. What can you do?

- Operate within a delivery time box focused on small batches of requirements of about equal size.
- Collaborate with your product owner and team to define a set of requirements that fits the team’s work capacity for each iteration.
- Work only on requirements (a.k.a. stories) that you’ve committed to deliver in the current iteration (or in preparation for the next iteration’s planning).
- As you elicit requirements, validate them by defining “doneness” criteria and user acceptance tests for each requirement.
- Create, or help your product owner create, requirements documentation that provides value to someone.
- Substitute user acceptance tests wherever possible for some, if not all, of your iteration-level requirements documentation
- Ensure that each requirement is ranked, and help your product owner regularly re-rank changing requirements.

Relearning

Software development is knowledge work composed of many continuous and compounding decisions. It’s difficult to remember small decisions, and their effects, as development

expands. To provide a knowledge trail, many organizations try to capture decisions in documents and processes. The risk is that they become pure bureaucracy—burdensome, meaningless, and wasteful. As an agile business analyst, what can you do?

- Conduct your analysis as close as possible to the time when it will be consumed, such as when code and tests are built.
- Challenge the business value, form, and format of each document and analysis artifact. Ask, “Do you really need that entire document? Why? When? What other ways can we satisfy that need? Work with audit, regulatory, and legal to define only the necessary and sufficient documentation.
- Help your team obtain product owner sign-off on each “done” requirement throughout the iteration.
- Conduct dependency analysis to prevent or mitigate rework resulting from undetected requirements dependencies.

Handovers

Handovers include moving artifacts from one team member to another. An example is a business analyst acting as translator between businesspeople and technical staff by creating requirements documents that then need to be signed off and handed off to technical staff.

What can you do?

- Act as an aide or coach, and not a translator, to your product owner (unless you are the explicitly designated product owner).
- Explore requirements directly with your product owner, subject matter experts, and the delivery team.
- Use low-fidelity analysis techniques (mockups, simulations, whiteboards and flip charts, role-playing) to increase the density of information exchange.
- Have developers and testers join you when you interact with users and stakeholders so they have a level of first-hand understanding of the problem.
- Sit with developers as they implement and testers as they test so you can clarify and define details precisely when they are needed.

Task Switching

When a developer or tester must seek out, e-mail, or schedule a meeting with a product owner or analyst to answer questions about a requirement, that is task switching. If you, as business analyst, hunt down a product owner to clarify business rules or check with the data administrator about the sources of a data item, you’re task switching. Working on multiple tasks in a short time saps your capacity to finish a single task and delays your completion of any of them. What can you do?

- Work on one project at a time or within one business domain.
- If you must multitask, cluster your time for thinking and acting so it is focused on the same domain.
- If you have rich domain expertise and understanding of business value, you may be the product owner, with direct decision-making authority about requirements priority and doneness.
- Be available to others so you do not block their work and force them to task switch.

Delays

Whenever someone or something is queued up and waiting to be started, you have waste: time is wasted, queued items age, and downstream resources are idle. What can you do?

- Make sure each requirement is ranked for priority with business value in mind.
- Encourage clear decision-making rules. At the start of each iteration, the product owner must decide which work is highest priority.
- Use collaborative techniques, such as informal workshops and prototypes, that allow stakeholders to “play” with requirements and make immediate decisions.
- Clarify requirements by focusing on story doneness and user acceptance testing.
- Show finished work right away to the product owner.
- Make sure that subsets of roughly equal-sized requirements are ready to be pulled into each iteration planning workshop to provide an even flow of work.

Defects

The later a software defect is detected, the more it costs to correct. Defects originating in requirements (missing, erroneous, conflicting, or unnecessary requirements) are costly. What can you do?

- Ensure that each story is small enough to work on in a single iteration, enabling defects to be found and fixed in smaller pieces and time scales. (This minimizes relearning, handovers, task switching and delays as well).
- Define doneness criteria for each requirement (story) as part of analysis before or during iteration planning.
- Create low-fidelity prototypes of stories to check for understanding and find missing requirements.
- Conduct research on ambiguous backlog items as they move up the backlog stack (I call these “analysis spikes”).

Agile + Lean = Added Value

As an agile business analyst, you are skilled at good thinking. With an eye toward waste, you can help deliver business value to your organization by adopting lean principles in your agile projects.

References

Gottesdiener, Ellen. *The Software Requirements Memory Jogger: A Pocket Guide to Help Software and Business Teams Develop and Manage Requirements*. GOAL/QPC, 2005.

Larman, Craig and Bas Vodde. *Scaling Lean & Agile Development: Thinking and Organizational Tools for Large-Scale Scrum*. Addison-Wesley, 2008.

Poppendieck, Mary, and Tom Poppendieck. *Agile Software Development: An Agile Toolkit*. Addison-Wesley, 2003.

Poppendieck, Mary, and Tom Poppendieck, *Implementing Lean Software Development: From Concept to Cash*. Addison-Wesley, 2007.

Shingo, Shigeo. *A Study of the Toyota Production System*. Productivity Press, 1981 (Japanese), 1989 (English).

Thanks!

The author would like to thank Tom Poppendieck for his review and incisive comments on a draft of this article.

Author Information

Ellen Gottesdiener, Principal Consultant, [EBG Consulting](#), helps you get the right requirements so your projects start smart and deliver the right product at the right time. Ellen's company provides high value training, facilitation, and consulting services to agile and traditional teams. An agile coach and trainer with a passion about agile requirements, she works with large, complex products and helps teams elicit just enough requirements to achieve iteration and product goals.

Ellen's book [Requirements by Collaboration: Workshops for Defining Needs](#) describes how to use multiple models to elicit requirements in collaborative workshops. Her most recent book, [The Software Requirements Memory Jogger](#) is the "go-to" industry guide for requirements good practices. In addition to providing training and consulting services and coaching agile teams, Ellen speaks at and advises for industry conferences, writes articles, and serves on the Expert

Review Board of the International Institute of Business Analysis (IIBA) Business Analysis Body of Knowledge™ (BABOK™).

You can subscribe to EBG Consulting's offers a free monthly eNewsletter "*Success with Requirements*" offering practical guidance and requirements-related news. When you sign up, you'll receive a free article on essentials for scoping your requirements.