Agile Business Analysis in Flow: The Work of the Agile Analyst (Part 2)

by Ellen Gottesdiener
EBG Consulting, Inc.: www.ebgconsulting.com

(This article first appeared in Modern Analyst, August 2009)

In Part 1 of "Business Analysis in Flow – The Work of the Agile Analyst ," I talked about the new skills and attitudes business analysts need to bring to agile development. When your organization adopts this value-centered approach, you need to have, as I wrote, "a tolerance for ambiguity along with a concurrent drive for specificity and closure."

Now it's time to talk specifics. What exactly do BAs *do* in agile development? How will your activities differ from those of traditional development? Let's take a look at agile business analysis from the perspective of the activities that make up requirements development and management, comparing traditional with agile analysis. .

Setting the stage: Requirements planning activities

To set the stage for requirements, the team strives to create a shared understanding of the product by all the stakeholders.

Traditional Analysis	Agile Analysis Adaptation
Attend project chartering sessions to define a vision, glossary, requirements risks, and product stakeholders.	● Design, facilitate, or participate in product vision and roadmapping workshops. ● Help your customer understand which roles and themes to best deliver in each product release. ● Help your customer and team identify logical groupings of value-based requirements, and use these groupings to create a product roadmap showing incrementally delivered requirements over time. These requirements often take the form of minimally marketable features, stories, or epics (i.e., large stories that cross releases), use cases (high level only), events, or a

	combination.
Review and modify a list of tasks, time, and delivery dates in a work breakdown structure plan developed by the project manager.	Design and facilitate (or participate in) release and iteration planning workshops. Regularly prune the product backlog by collaborating with team members to generate a relative size estimate for backlog items. Conduct analysis "spikes" (short, timeboxed research stories) to elaborate on backlog items that need more analysis, researching requirements and their priorities.
Generate a SWAG ("S#*&-Wild-Ass-Guess") estimate of time, effort, or cost for each requirement in the specification or user requirements document.	During iteration planning, together with the rest of the team, write down the needed tasks to deliver each user story, and estimate how many hours they will take. Share actual time usage information with your team so that the team can track progress via visual graphs ("information radars") such as burndown, burn up, or cumulative flow diagrams.

Requirements elicitation activities

During requirements elicitation, the team identifies the sources of requirements and then discovers, derives, evokes, and elicits requirements from those sources.

Traditional Analysis	Agile Analysis Adaptation
Plan how to elicit requirements using a variety of techniques.	• Use face-to-face, collaborative elicitation techniques (workshops, prototypes) as much as possible while avoiding techniques (interviews, surveys, documentation study) that require longer lapse times or interpretation.
Plan, design, and facilitate requirements workshops over weeks (or months).	Plan and facilitate short, informal requirements modeling sessions throughout each iteration. Plan and facilitate product vision and roadmapping workshops and release planning workshops. Teach your customer about supplemental analysis models so that they can question, participate, critique, review, and approve them (this should be done in traditional

	projects as well). - Sketch out prototypes and identify user acceptance test data in real time, while a story is being designed, coded, and prepared for testing.

Requirements analysis activities

During analysis, the team seeks to understand and define requirements so that stakeholders can prioritize their needs and decide which requirements to build.

Traditional Analysis	Agile Analysis Adaptation
Define the scope up front by using a set of requirements models as the basis for detailed modeling.	- Help your customer define the vision and the scope up front—at a high level only. - Help your customer and team create lightweight models during product roadmapping and release planning. These models help customers carve out a value-based release schedule that balances business priority with architectural dependencies. - Collaborate with architects and developers on design to ensure that requirements include the technical aspects of the product.
Develop analysis models for the entire set of requirements that are in scope.	- Help your customer and team develop stories (user stories as well as stories that incorporate or separately define quality attributes). - Help your customer and team develop and extend analysis models that support understanding backlog items selected for delivery in an iteration—if and when needed.
Ask the customer to prioritize requirements using a ranking scheme. If the customer is not available, do the ranking yourself.	- Help your customer assign a business value and a ranking to each backlog item. - Help your customer understand requirements dependencies that might warrant adjustments to backlog rankings. - Question rankings based on goals or themes for upcoming release or iterations. - Assist your customer and team to right-size high-priority backlog items that are too big to deliver in combination with other high-priority backlog items in the next iteration.

Requirements specification activities

Specification involves refining and organizing requirements into documentation (typically a software requirements specification). This includes the entire set of functional and nonfunctional requirements to be transformed into design, code, and tests.

Traditional Analysis	Agile Analysis Adaptation
Write a requirements specification.	● Help your customer and team write stories (or if you're acting as proxy customer, you write them). ● Create doneness criteria for stories so that each becomes a well-defined, small piece of valuable software for delivery in the next (or current) iteration. ● Create user acceptance tests or sample input and output data for each story. ● Determine the form and format of documentation that is necessary and sufficient for requirements-related work-in-progress, handover, or product documentation.

Requirements validation activities

During validation, the team assesses whether the product satisfies user needs and conforms to the requirements.

Traditional Analysis	Agile Analysis Adaptation
Set up and run meetings to review and sign off on requirements documents, and help customers run acceptance tests after the entire product's code has been created.	● Meet with the customer and some team members to prune the backlog (once or twice each week). ● Participate in iteration demonstrations and listen to stakeholder feedback on the delivered requirements to learn the customer's real needs and determine how to adapt the evolving product. ● Plan and facilitate, or participate in, iteration retrospectives, and learn from the customer how you can help deliver value faster.
Communicate with developers or testers (or respond to their e-mails and calls) to explain information in	● Conduct just-in-time analysis modeling with customers and your team to validate the business value of each story and to

the requirements document; attend or run formal requirements review meetings.	ensure it will be delivered to the customer's satisfaction. ● Participate in daily stand-ups. ● Sit with developers and testers as they are building code and tests to explain the story and its doneness criteria.
Help testers create user acceptance tests, or run those tests, after the entire product has been designed, coded, and unit/system/integration tested.	● Define input data and expected results or specific user acceptance tests as part of defining doneness for each user story, iteration by iteration.

Requirements management activities

Requirements management involves monitoring the status of requirements and controlling changes to the requirements baseline ("a point-in-time view of the requirements that have been reviewed and agreed upon to serve as the basis for further development," Gottesdiener 2005).

Traditional Analysis	Agile Analysis Adaptation
Establish the requirements baseline, document change control processes, and generate requirements trace matrices.	● Help the customer and team establish a product backlog and define the smallest necessary requirements attributes for each backlog item. ● Help the customer and team define "just enough" requirements tracing needed to satisfy external regulatory body expectations. ● Help the team determine simple, meaningful requirements mapping and organizing (features to stories, events to stories, etc.). ● Define simple, unobtrusive ways to trace stories, with the aim of capturing metrics that will be useful for reuse and promoting development efficiencies.
Attend or schedule change control meetings.	● Help the customer and team prune the product backlog continually (reprioritize items, break down stories, assign rankings, estimate size, and explore requirements dependencies that will impact architecture

	and therefore release planning). - Help the customer maintain the product backlog items (on story cards on a wall, in a spreadsheet, or using an industrial strength agile requirements management tool)—or do this on behalf of the customer.

Learning: The heart of agile success

A mantra for agile teams is "inspect and adapt." This means regularly checking on the delivered product and the processes used. Continuous improvement (called "kaizen" in lean approaches) is essential to agile success. How do you inspect and adapt your business analysis work to learn and develop?

Traditional Analysis	Agile Analysis Adaptation
- Participate in milestone or project "lessons learned" sessions to find out what went wrong, what went right, and who is responsible for the problems. The project manager fills out the lessons learned template and writes the closeout document. - Sit with your manager once or twice a year for a performance review, and get feedback on your performance, months or weeks later. Sometimes that feedback includes second-hand comments from your customer and team.	- Use acceptance tests, examples, sketches, simple drawings, and face-to-face communication to get feedback on your understanding of requirements. - Participate in daily stand-up status meetings to hear the impact you are having on other people's ability to deliver. - On any given day, as an item you committed to deliver is deemed done, show it to the customer to get feedback on it and confirm that the conditions of satisfaction have been met. - Design and facilitate, or participate in, iteration and release retrospectives (every two or three weeks, depending on your iteration timebox) to learn what works, learn what to adapt, and collaboratively agree on one or two things to do differently in the next iteration or release. The goal is to learn, adapt, get better, and experience joy in your work.

The new world of agile analysis

So there you have it – a bird's-eye view of how business analysts operate and add value in agile projects. As you can see, this approach calls on you to stretch your analysis muscles.

As an agile analyst, you are deeply committed to delivering business value and building the right product as soon as possible. As a member of an agile team, you are less concerned with roles and job boundaries, and more concerned with delivering as a team.

You experience the rhythm of successive elaboration and product delivery. You thrive on feedback and small, continual improvements. What's more, you have an intense need to self-reflect, communicate transparently, improve your skills and abilities, and serve your team and customer. You thrive on the energy and joy of being in rhythm with an agile team.

References

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

Resources and Readings

Video: A brief overview of Agile Business Analyst
Additional Readings and Resources

Thanks!

The author would like to thank Phil Abernathy, Susan Block, Mary Gorman, Kamal Singh, Norman Stang, and Stephanie Weiss for their helpful review and feedback on a draft of this article.

Author: 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. You can follow Ellen on Twitter or contact her via email.