

Agile Requirements by Collaboration: Making Smart Choices About What and When to Build[©]

by Ellen Gottesdiener

EBG Consulting, Inc.: www.ebgconsulting.com

Copyright © EBG Consulting, Inc., 2009

(This article first appeared in InformIT, August 24, 2009 issue)

How do you make smart choices about what to build, and when, on an Agile team that values interaction and customer collaboration? Ellen Gottesdiener describes incorporating Agile requirements modeling into collaborative planning workshops.

On an Agile team, you need to make tough choices about requirements—collaboratively—and include the perspectives of all the product's business and technical stakeholders: the business sponsor, customers, users, testers, developers, architects, analysts, user experience designers, marketers, post-implementation supporters, trainers, and so on. What's more, you want to collaborate throughout the entire project in an efficient and consistent manner.

Collaboration yields a healthy project community sharing focus, values, and trust. Collaborative workshops provide an effective venue for Agile teams to work together transparently to make the complex decisions about what to build, and when. By *collaborative workshops* (or simply *workshops*), I mean structured meetings in which a carefully selected group of stakeholders—including content experts, product owners and customers, and the delivery team—work together to define, create, refine, and reach closure on deliverables.

Agile Planning Workshops

On an Agile project, requirements unfold within the context of the rhythm of Agile planning: product, release, and iteration (or sprint). You hold different kinds of workshops (or planning meetings) at different points in your project:

- Product roadmapping workshop
- Release planning workshop
- Iteration planning workshop

On Agile projects, planning and requirements elaboration converge. Planning workshops incorporate requirements *exploration* as well as *allocation*. Exploration involves eliciting, elaborating, and analyzing requirements (requirements modeling) for that planning horizon. This design allows you to allocate portions of the requirements to the next appropriate planning horizon.

You calibrate your requirements focus based on the applicable workshop:

- During product roadmapping workshops, you explore and allocate the *big-view* of requirements to map out a strategy for the entire product.
- In release planning workshops, you focus on a smaller time horizon to get a *pre-view* of requirements for the next release.
- In iteration planning workshops, you explore and plan for a small, concise set of requirements for the immediate sprint—the *now-view*.

These planning workshops involve the project and product community—the technical team and the business customers (product owners). Typically, participation in the product and release workshops involves people at higher levels in the organization.

Agile Requirements Modeling in Planning Workshops

Requirements workshops have always had elements of agility. These workshops build and sustain collaboration among technical and business people, generate shared understanding and mutual trust, incorporate hard-nosed assessment of project tradeoffs, and strive to uncover just enough information to deliver value to all stakeholders.

You adapt requirements workshops on an Agile project by incorporating requirements exploration (modeling) within Agile planning sessions. At times, these modeling activities are more like workshops, in that they use a skilled facilitator who has thought ahead of time about how to orchestrate the participants' interactions to optimize outcomes. At other times, the modeling is more informal and ad hoc. In any case, requirements modeling is focused on the necessary and sufficient requirements for the requirements focus at hand: big-view, pre-view, or now-view.

Let's explore the Agile requirements modeling activities to incorporate at each level of Agile planning.

Product Roadmap Workshop

The *product roadmap* workshop is integral to developing and articulating your company's product strategy (see Table 1). Before creating your product roadmap, you need to understand your market and review your technology capabilities and trends. When your organization is ready to allocate and align people and money with a product development effort, it's time to conduct a product roadmap workshop.

Alternatively, you can conduct a product roadmap workshop for a team in flow that doesn't yet have a product roadmap. How does this happen? Some Agile teams begin their journey by getting into a rhythm of delivery (it usually takes three or four iterations). They have a product backlog (a list of work to be done) and a mental sketch of the product, but for only one release. Lacking an overall product strategy or vision, they begin to question the sequence of delivery and become concerned about the sustainability of the architecture. They're ready for—and need—a product roadmap.

The key requirements-related deliverables for your product roadmap workshop are the vision statement and the product roadmap. You also incorporate product management deliverables, such as market segments and competitive positioning, to help solidify your product strategy.

The product roadmap identifies product themes and the cohesive chunks of valuable features that the team needs to deliver in each time slice. These time slices will become releases in the product roadmap. The workshop's deliverables also feed the product backlog.

Inputs to a successful product roadmap workshop include an analysis of market needs and trends, competitive offerings, and critical time-based occurrences; the organization's business strategy; and technological, financial, and time constraints.

Table 1: Product Roadmap Workshop: The Essentials

Workshop Purpose	Sample Requirements Deliverables	Participants	Timing
Define the product's vision and scope. Identify the features to deliver for the entire product.	<ul style="list-style-type: none"> • Vision statement • Product roadmap • List of 	<ul style="list-style-type: none"> • Project sponsor • Executive advisers from marketing and the 	<ul style="list-style-type: none"> • Once per product, revised periodically • <i>Time</i>

Develop a product backlog.	stakeholders, including customers who buy or authorize spending and users (personas) <ul style="list-style-type: none"> • Themes • Features or feature levels, including nonfunctional requirements • Context diagram • Events, use cases, "epics," or stories 	organization <ul style="list-style-type: none"> • Strategic and tactical product owners or champions • The delivery team (or subset of senior team members, especially those with domain and technical architecture expertise) 	<i>horizon: 1–3 years</i>
----------------------------	--	--	---------------------------

Release Planning Workshop

Teams benefit from having a time horizon with a narrower focus than the entire product. At the same time, teams benefit from a wider view than just the immediate iteration (sprint). A release planning workshop provides this perspective (see Table 2). You conduct this workshop for each release, roughly every 1–4 months.

Your key requirements-related challenges in a release planning workshop are twofold. First, you need to identify the requirements that deliver an overarching theme to well-understood users (personas). Second, you focus on balancing business priorities with architectural dependencies.

NOTE

A persona is an archetype representing a composite of similar end users. Personas are described as if they were real people with motivations, needs, and behavior patterns, in order to help the team understand usage requirements.

With an eye toward requirements, inputs to a successful release planning workshop include the product roadmap, a definition of team capacity and velocity, an analysis of the state of the product architecture, an awareness of requirements dependencies, customer priorities, the release date, and a working set of "doneness" criteria for the release.

Table 2: Release Planning Workshop: The Essentials

Workshop Purpose	Sample Requirements Deliverables	Participants	Timing
Identify what to deliver for a given release. Elaborate the product backlog; prioritize, estimate (at a high level), and prune the backlog.	<ul style="list-style-type: none"> • Release theme and release acceptance criteria • User roles or personas • Epics, users stories, use cases, story maps, or a combination • Conceptual data or class model • Quality attributes and interface stories (or how the team accounts for them) • Supplemental analysis models, as needed, to explore release-level requirements 	<ul style="list-style-type: none"> • Strategic and tactical product owners or champions • The entire delivery team 	<ul style="list-style-type: none"> • Once per release • <i>Time horizon:</i> every 1–4 months

Iteration Planning Workshop

The timing of iteration-level requirements modeling activities is not as cut-and-dried as for other Agile planning workshops. You conduct requirements modeling as part of work-ahead planning for the iteration, as part of the iteration planning workshop itself, and throughout the iteration (see Table 3).

Table 3: Timing for Iteration-Level Requirements Modeling

Iteration Requirements Modeling	Purpose
Work-ahead (during the prior iteration)	Select and prioritize backlog requirements that are likely to be chosen for the next iteration, size those requirements, and learn what you don't know about them so further analysis can be done prior to iteration planning.
Incorporated within iteration planning	Explore requirements details so you can estimate, task, and make a commitment for delivering a set of requirements for the iteration.
Throughout the iteration	Analyze requirements details as necessary so you have sufficient requirements knowledge to build, test, and demonstrate the requirements within the iteration.

Before your iteration planning workshop, use iteration requirements modeling sessions to do some *work-ahead* analysis. This enables the team to have a productive and efficient planning session. During these work-ahead sessions, the team (or a subset of the team) will explore requirements for backlog items likely to be delivered in the next iteration, identify requirements details (for example, user acceptance tests, data attributes and their sources, external interfaces, quality attributes, business rules), determine which stories might need more research, explore requirements dependencies, and prune the product backlog.

In your iteration planning workshop, the team estimates, plans, and commits to delivering a set of backlog items (usually in the form of user stories). Your planning workshop incorporates short bursts of requirements exploration (that is, lightweight requirements modeling activities) so you can learn enough about the requirements to gauge the tasks and time each item will take. Your key requirements-related deliverable are "right-sized" stories—

small, concise user requirements with clear conditions of satisfaction that fulfill the iteration's theme or goal.

NOTE

A *user story* is a concise, sharply defined user requirement that briefly describes something valuable the user needs to accomplish. The most common format for writing a user story is "As a <user role | persona> I want to <action | behavior> so that <business value>."

Inputs to successful requirements modeling during iteration planning are the prioritized backlog, an estimate of team capacity and velocity, an understanding of the state of product architecture, an awareness of requirements dependencies, the iteration delivery date, and a working set of "doneness" criteria for the iteration. Table 4 summarizes the requirements-related essentials.

Table 4: Iteration Planning Workshop: The Essentials

Workshop Purpose	Sample Requirements Deliverables	Participants	Timing
Identify what to deliver for a given iteration	<ul style="list-style-type: none"> • Right-sized user stories • Quality attributes and interface stories (or an analysis of how they're accounted for in user stories) • User acceptance tests • Supplemental analysis models, as needed, to explore requirements for iteration stories (for example, 	<ul style="list-style-type: none"> • Tactical product owner or champions • The entire delivery team 	<ul style="list-style-type: none"> • Two or more times before iteration planning workshop for work-ahead • During iteration planning • Multiple times throughout the iteration • <i>Time horizon:</i> every 2–3 weeks

	refactored data or class model, user story maps, state diagram)		
--	---	--	--

During iteration planning, you tolerate some necessary incompleteness, deferring some details about what you'll test and code until you actually do that development work. Then, "just in time," you have conversations about the requirements you're building and testing, and conduct short requirements modeling sessions (what Scott Ambler calls "[model storming](#)").

At times, these modeling sessions warrant more formality. Perhaps the right stakeholders are unavailable. Maybe requirements details are unclear and need more in-depth analysis, or the requirements are complex and interconnected. In those instances, you schedule the modeling session and optimize time by planning a series of collaborative requirements modeling activities.

For example, a team member will step in as requirements facilitator and investigate key user roles or personas. She creates a lightweight profile of the persona's motivations and needs, targeting the current iteration. The facilitator then schedules a modeling session and kicks it off by having the team and relevant stakeholders review and revise the persona's profile.

Next, the facilitator leads the group through a listing and validation activity; they name all the iteration's stories and crosscheck them against the iteration theme and persona's needs. This activity often surfaces questions about the scope of the stories, which are resolved. Then the facilitator leads the group through modeling activities to specify the necessary data, state, and rules for stories. This approach reveals conflicts and overlaps in the stories, allows the product owner (customer) to clarify acceptance criteria, and exposes potential reuse of user acceptance tests.

By incorporating a light touch of forethought and pre-work, as well as using a skilled facilitator, the team saves time. Without this session, throughout the iteration team members' time is sapped by multiple interruptions to discuss requirements ambiguities and inconsistencies.

Regardless of whether you're conducting requirements as work-ahead, during iteration planning, or throughout the iteration itself, these Agile requirements sessions focus *only* on the requirements for your current (or next) iteration. They're informal (using whiteboards, flipchart modeling, sticky notes, and other low-fidelity tools) and require less pre-work than product roadmap or release planning workshops. Yet these sessions benefit from many of the

patterns, design guidelines, and collaboration techniques I described in [Requirements by Collaboration: Workshops for Defining Needs](#).

Agile Requirements Workshops in Context

Success with Agile development depends on delivering the right product requirements at the right time. Agile teams constantly plan and re-plan to make those decisions, attempting to optimize business value at regular intervals within the constraints of time, money, and people.

Decisions about requirements are crucial to making smart delivery choices. The team must agree on the optimal set of requirements to deliver next. Agile planning should balance customer needs with the architectural impact of requirements dependencies, incorporating the implications of both functional and nonfunctional requirements on the architecture and customer. Successful Agile teams collaborate to make these decisions.

At the same time, Agile teams need to tune the breadth and depth of their requirements exploration to the cadence of Agile delivery. Using Agile requirements modeling interwoven into Agile planning workshops, teams effectively define sound delivery decisions and feed the engine that makes Agile teams run—collaboration.

NOTE OF APPRECIATION

The author would like to thank Nanette Brown, Mike Cohn, Mike Cottmeyer, Rachel Davies, Mary Gorman, and Scott Sehlhorst for their helpful review and feedback on a draft of this article.

Additional Reading

- Scott Ambler, "[Model Storming: An Agile Best Practice](#)"
- Jane Cleland-Huang and Mark Denne, [Software by Numbers: Low-Risk, High-Return Development](#)
- Mike Cohn, [Agile Estimating and Planning](#)
- Ellen Gottesdiener, "[Agile Requirements, in Context](#)"
- Ellen Gottesdiener, [Requirements by Collaboration: Workshops for Defining Needs](#)
- [Luke Hohmann](#), [Steve Johnson](#), and Rich Mironov, *Living in an Agile World: The Role of Product Management When Development Goes Agile* (Kindle book, 2009, available on Amazon.com)

- Hubert Smits, "[Five Levels of Agile Planning: From Enterprise Product Vision to Team Stand-Up](#)"
- Jean Tabaka, [Collaboration Explained: Facilitation Skills for Software Project Leaders](#)

Author

[Ellen Gottesdiener](#) is the founder of EBG Consulting, which helps business and technical teams build a shared understanding of product needs so they deliver value at the right time. She is the author of [Requirements by Collaboration: Workshops for Defining Needs](#) and [The Software Requirements Memory Jogger](#). Ellen is an internationally recognized facilitator, trainer, speaker, and expert on requirements development, product chartering, retrospectives, and collaborative workshops. Subscribe to EBG Consulting's free monthly newsletter, "[Success with Requirements](#)," and visit the [EBG website](#) for a variety of resources. You can reach Ellen at ellen@ebgconsulting.com.