Specifying Requirements With a Wall of Wonder

by **Ellen Gottesdiener**
Principal, EBG Consulting

Most of us are familiar with the tension between some developers' desire for loose, flexible processes and project and business demands for rigorous, disciplined software development. In my experience, successful companies are those that assemble collaborative teams of dedicated software professionals who exploit the strengths of both positions. These people recognize that nuggets of rigor and discipline lie at the heart of well-executed agile methods, and that within the soul of disciplined methods lie adaptability and open-mindedness. In project after project, I have seen those with a collaborative vision and approach succeed where traditional, "lone wolf" approaches have failed.

Collaborate *comes from the Latin words* laborare *(to work) and* com *(with), and so literally means* to work together. *Collaboration occurs when a group of people with a common and well-defined goal integrate their individual knowledge and skills to deliver on that goal. When I facilitate for development teams, I conduct collaborative workshops in which we create requirements specifications and related deliverables such as use cases, business rules, data and class models, and plans. The workshops comprise a healthy mix of business people (users and customers) and IT (information technology) professionals, all driven by a single goal: to deliver the best software solution to meet a business need.*[1] *The processes and techniques we use in these workshops contribute to building healthy team relationships, trust, and shared meaning.*

In this article, I describe one very effective collaborative technique -- the Wall of Wonder (WoW) -- which helps software teams produce the kind of detailed, sharply-defined requirements that effectively guide development. As an "emergent" deliverable, requirements evolve through exploration and examination using representative forms such as models and

prototypes. A collaborative approach allows business and IT specialists to explore their requirements through these means, while accommodating the necessary fluidity of the requirements process.

Using the Wall of Wonder

Collaboration requires shared space in which the deliverables can emerge, and the simplest -- and cheapest -- form of shared space is a wall. Walls have been used by generations of human societies to record events and tell stories (this spans the entire range of human history, from cave paintings to modern graffiti), and to express hopes and wishes. The Wall of Wonder (WoW) is both a communication medium and a storage place for the tools and materials (which we'll discuss below) and the content of the collaborative effort.

To make best use of the wall, groups need a structured process -- a fast, flexible way to leverage the wall space and engage in healthy interaction. Dynamics for healthy groups iterate around a well-recognized cycle first described by B.W. Tuckman,[2] who labeled the phases *forming*, *storming*, *norming*, and *performing*. *Forming* involves finding common goals and purpose. During the *storming* phase, members engage in open disagreement, which, under healthy circumstances, strengthens the group and promotes deeper understanding. *Norming* is the process of finding commonly acceptable ways for all group members to interact. In the *performing* phase, the group becomes task-oriented and focuses on producing the agreed-upon deliverables. Although you can't take shortcuts within this nonlinear set of dynamics if you want usable results, you can at least accelerate it. Certainly, a group that has worked together previously, and whose members are familiar with each other before the WoW session, may begin norming and performing right away. To produce a deliverable, however, they will still need to follow all the steps described below.

Storyboarding is one kind of wall work that follows the cycle Tuckman describes. A storyboard is a series of continuing panels, sketches, or scenes depicting a plot or sequence of actions. In business, storyboards are popular for solving problems and creating collaborative plans. My Wall of Wonder approach is based on a storyboarding technique I learned from the Institute of Cultural Affairs.[3] The group uses text or diagrams to build requirements, an iteration plan, or other important deliverables by successively using individual, subgroup, and whole group activities to generate items such as desired project features, business rules, use cases, data elements, screen navigations, and so on. I call it a Wall of Wonder (WoW) because of the wondrous results groups can achieve. Table 1 outlines the rationale for this approach, and Figure 1 shows a step-by-step flow.[4] For a more detailed explanation of the steps, see the Sidebar.

Table 1. Collaboration Pattern: Wall of Wonder (WoW)

Name	***Wall Of Wonder (WoW)***
Context	A group of stakeholders needs to solve a problem or create a deliverable such as a model or plan. A same time/same place group meeting seems to be a good choice because it will speed the process.
Problem	Unstructured group meetings are not always the best way for participants to interact. If there are introverts, more vocal individuals, or more organizationally powerful people present, some valuable input may be lost.
Solution	• In a group meeting, allow time for individual thinking, followed by small group work to combine individual ideas and then whole group work on the wall. • Begin with a clear focus question. • Have all work visible on the wall. • Permit the group to logically arrange the elements on the wall • Respect individual time and the need to think alone and then post the result on the wall. • Establish a pattern of individual -> triad -> whole group -> individual. • For longer collaborative events, rotate membership in subgroups.
Consequences	Integrating multiple perspectives produces a higher-quality product than one produced by a subset of individuals. Greater team collaboration and goodwill are established because the process allows participants to think both alone and together. As a result, teams can create complex requirements, models, plans, or structures in a short time (one hour to one day).
Entry Criteria	• Knowledgeable participants representing all key IT and business perspectives who are sharing the same time and place • Understanding of what deliverables are to be created and what is required to create them • Focus questions • Room(s) with sufficient wall space • Low-tech tools such as post-its or sticky wall with large cards; or high-tech tool such as an outliner or drawing tool along with a technographer

Exit Criteria	All deliverables are visible on one or more walls; the deliverables contain both detailed and summary items, are logically cohesive, and are understood and agreed upon by all participants.
Uses	Define use cases by release, determine the scope of events or use cases, define a data or class model, create an actor (role) map, specify business rules in a template format, associate business rules with use cases, define goals and objectives, design an organization, create a communication plan, generate a project, phase, or iteration plan, define selection criteria for a software package, design a low-tech user interface prototype, specify use case steps.

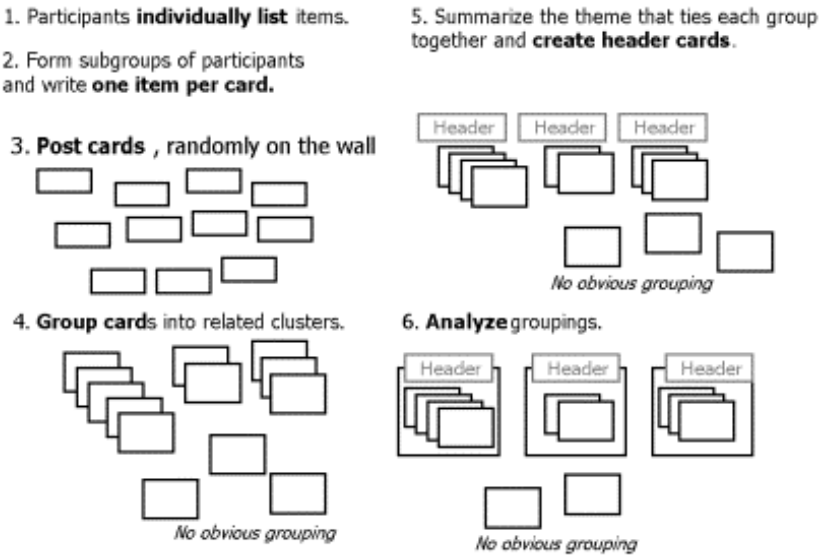

Figure 1: Wall of Wonder Process

At the beginning of each WoW activity, a neutral facilitator poses focus questions to trigger responses from participants. (The focus questions vary, depending on the purpose; we'll look at examples later). After people generate detailed items on their own, subgroups write agreed-upon items on detail cards or sticky notes. Here's where the wall comes into play. These cards are then displayed on the wall for the whole group to focus on. When everyone reaches agreement on the posted items, the cards are grouped under header cards (usually in a contrasting color from the detail cards). At the end of the process, the work of the whole group is displayed on a single, large wall.

Three Wall Strategies

In the bottom-up approach shown in Figure 1, participants begin with detailed items, and then work their way "up" to higher-level categories, designated with header cards. Along the way, they sort and eliminate items, including project features, business rules, use case steps, and data elements.

In a top-down approach, participants prepare the header cards before the workshop or in an earlier collaborative activity. Starting with these broad categories, participants then work their way "down" by supplying lower-level details. This approach is useful when the categories are well known and accepted by participants ahead of time. Examples of high-level categories or header cards might be project phases, SWOT (strength, weaknesses, opportunities, threats) analysis categories, use-case names, business rule sets, named data entities or business objects, or object lifecycle phases.

In a middle-out approach, participants generate items more randomly, without regard to categories. Then team members sort the items, specify more detailed items, and converge on the categories. An example might be asking participants to list use cases. After team members sort, combine, add, and clarify the cards, something that originally was posted as a use case may turn out to be a use-case step, a set of use cases, or a use case that is out of scope.

Group Size and Balanced Participation

At a WoW session, your goal is to foster healthy group participation, achieving a state in which, as Aristotle put it, "The whole is more than the sum of its parts; the part is more than a fraction of the whole." There is a delicate balance between group size and getting the right participants ready to work the wall.

When a group includes people with different knowledge, perspectives, and backgrounds -- as is often the case in requirements workshops - then communication is complex. That is one reason subgroups are useful. You'll want to form subgroups if you

Steps for a WoW Session

1. **Make individual listings.**
   a) Briefly explain the overall flow to prepare participants. Make sure they have paper or cards, and then ask the focus question. (Remember, the focus question sets the content for the Wall of Wonder and is critical to the process).
   b) Ask the focus question. Allow people to think and brainstorm alone for five to seven minutes, listing their answers on the card or paper.
   c) Ask participants to mark their top three items, thereby sorting those they consider most important, relevant, useful, or critical. I prefer to give them a neutral criterion for sorting by saying, "Draw a star next to your top three items."
2. **Assign one item per card, in subgroups.**
   Divide the group into subgroups. Determine the subgroups ahead of time or form them at this point. The task of the subgroups is to arrive at a list of items in response to the focus question by sharing, merging, converging, and adding to their cards. Shoot for seven to nine items, creating a card for each. I usually have the group choose a *leader* to keep them on task and on time and a *recorder* to do the recording for the group. I instruct the recorders to:
   - Use a black marker to write one item per card.
   - Describe the item with three to five words (cut to the chase).
   - Write big.
3. **Post cards on the wall.**
   Reconvene as a whole group. Ask for one card per group using questions such as "Which card was most obvious to your subgroup?" "Which card did you select the quickest?" or "Which is your most controversial card?" As you place each card on the wall, read it *exactly as it is written*

have more than four participants. This is critical for preparing for the "plenary" part of the process (Steps 3-6 in Figure 1). In later steps, the whole group reconvenes to model on the wall, working with potentially many elements.

Subgroups should have no more than three people; this small number allows them to get the work done quickly, encourages everyone to contribute, and discourages one-on-one conversations that can eat up time. Of course, dyads (groups of two) may be even faster, especially at the beginning of a workshop, because they allow the two people to get comfortable with each other. If you have more than six or eight people in your workshop, however, dyads may not be feasible.

Requirements Example: Creating an Actor Map

An efficient way to discover use cases is by using *actor maps*. An actor map (also called a *role map* or *actor hierarchy*) is a diagram depicting how actors relate to each other in terms of shared goals for interacting with the system (see Figure 2). The same actor can appear on multiple actor maps.

and ask the group if they have clarifying questions. If there are any questions or puzzled looks, ask the team members who wrote the card to explain its meaning and intent. Continue this process until all detail cards are posted.

4. **Group cards.**
   As this process unfolds, guide the team in grouping the cards. (Often, participants intuitively start to tell you to do this. If they don't, ask them whether anything on the wall "goes together.") Moving the cards, form columns or clusters, depending on the deliverable. Create new cards whenever a new item arises, when an item needs to be clarified, or when header cards are suggested.

5. **Create header cards.**
   Create a header card for all groups or clusters. If you were writing a header card for data elements in a data model, for example, the header card would be the name of the data entity or business object to which the data element belongs. If you already have header cards from an earlier step, then ask participants to reconsider the label and ensure that it is still relevant and meaningful.

6. **Analyze groupings.**
   a) Extend the content of the wall by asking everyone to review his or her original list. As they do this, ask them whether any cards should be added.
   b) Quality check the end product (see Table 2 for examples), and then use your decision rule process to reach closure.[5]

Note: As a follow-up to creating the Wall of Wonder, consider conducting a structured discussion of the wall's contents. Ask the group to reflect on both the product they created (and its implications for the project) and on the process they used to create it. Pausing to debrief -- to process both your product and your process - offers rich learning opportunities and is the mark of a truly collaborative team.

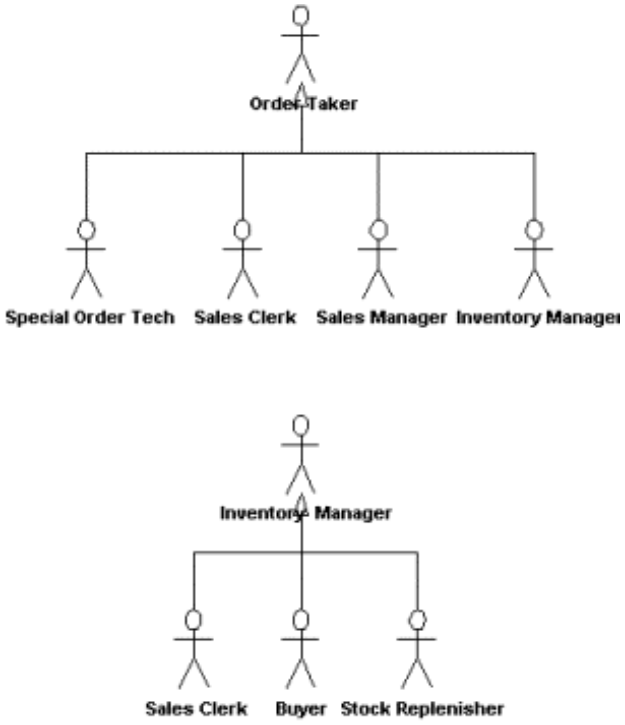

Figure 2: Examples of Actor Maps

To generate an actor map using the WoW technique, begin by asking participants to develop a list of actors (a bottom-up wall strategy). Write the actor names on cards, and then arrange them on the wall to form hierarchies. If the same actor appears in multiple hierarchies or maps, then create another card for that actor (see Table 2 for complete instructions).

Table 2: Wall of Wonder Collaboration Pattern: Defining Actors

General Steps	Brief Description	Potential Step Inputs	Focus Questions and Tips
1. Name the actors.	Use the focus questions to list all potential actors. First, have each person create a list on his or her own.	• Stakeholder classes,[6] including direct users (actors), customers, suppliers, and tertiary users • List of project goals and objectives from the charter • Context diagram[7] • Event table[8] • Organization chart • List of use cases	Who directly interacts with the system we're building? Which people and things in our context diagram directly give or get things from the system? What are the subject and object names from our event list? What people or things, such as databases or other systems, have we not yet identified that interact with the system? Given our project goals, who or what

			else will need to interface directly with the system? What other systems, databases, or external feeds need to interface with system?
2. List actors on cards in subgroups.	Form groups to collect and document their respective actors, one per card.	• List of actors generated in prior step	
3. Place cards from each participant group on the wall.	As a whole group, post cards on the wall one at a time, discussing and agreeing on the name for each actor.	• Cards from each subgroup's work Which card represents the person your group thinks is most critical to achieving our project goals?	Which card represents more of a supporting role with respect to our goals? Which card represents a role you had some debate about? Do you have a card to give me for posting that represents a system or a database?
4. Create hierarchies for shared actor roles.	As a whole group, consider each actor in turn. Use the focus questions to move actors into hierarchies, creating duplicate cards as needed for actors that appear on multiple hierarchies.	• Shared actor roles (generalized actors) from the preceding step	What are the roles we've identified? Which actors belong here? [Cluster the cards together.] Which actors belong there? Do we need to make a duplicate card for this actor? Are there other actors for this role? Where should we put the actors who haven't yet been put into a hierarchy?
5.Summarize and review the hierarchies.	Look for ways to combine, expand, or contract hierarchies to simplify the business.	• All information on the wall	Inspect each of the actor hierarchies. Which actors don't belong? What do the actors in each hierarchy have in common? Do we have all the actors who could play this role? Will we want other actors to also play this role? Are there actors who really should not be playing this role?

6. Finalize the actor map. a) QA the actor map against other models such as the event table, context diagram, or list of use cases b) Reach closure.	a) Ensure that each event has at least one actor who handles the event by initiating a system interaction. Ensure that each actor handles one or more events. Ensure that each actor is depicted on the context diagram. Revise as needed. b) Use the decision rule and decision rule process (see *Software Development Magazine*, January, 2001) to reach closure.	• Event table • Context diagram • Organization chart • Stakeholder classes • Use cases • Organization chart • Actor map on the wall • Your decision rules and agreed upon decision rule process	Devise questions that allow you to crosscheck the model. For example, if you have a list of events, ask: Which event(s) does each actor handle? Is there an actor to handle each event? Are there any unnecessary events-ones with no corresponding actor?

Notes for Table 2: Variations and Considerations

- Conduct the entire process as a whole group, particularly if the group consists of seven or fewer people.
- Prepare a draft of an actor map ahead of time to show the participants.
- Identify your subgroups before starting the activity, and decide which groups will get which input model (e.g., events) to work with when generating actors.
- After all the actors are posted on the wall, assign each one to a subgroup; members of this subgroup then form hierarchies independently. Then review the entire set as a whole group.
- After all the actors are posted on the wall, ask half of the participants to rearrange them into hierarchies while the other half takes a break. Then switch so that the people who are back from their break QA the first group's hierarchies, and give the first group their break. Finally, reconvene the whole group to walk through the actor map.
- Avoid using company titles for actor names.
- Suggest naming actors with an "er" at the end, such as "buyer," "stock re-supplier," or "customer query answerer."
- If you have taken a "middle-out" approach and already have a candidate list of use cases (actor goals), then assign a use case to each actor on the actor map, adding use case cards to the map.

I like to test the actor map (or any other deliverable) for "doneness." The doneness test is an agreed-on set of criteria that test whether the deliverable is *good enough,* and thus whether the group has reached closure. To test an actor map, I often combine a quality checklist with other models --- a list of events, stakeholder classes, or use cases, for example -- to crosscheck the deliverable for correctness, completeness, and consistency. At the end of this process, I ask the group to decide whether the actor map is done by using a collaboration pattern for decision making.[9]

WoW Warm-Ups

You can jump-start a requirements WoW session with a short imaging activity that helps participants concentrate on what will be important to put on the wall, and a brief mini-tutorial with an example, explaining the requirements' purpose. One essential warm-up is to use *focus questions* that direct participants' attention to specific items you want to get onto the wall. When the goal is to produce a requirements model, focus questions help direct the group's thinking toward an element of that model. They are also helpful to groups developing an iteration plan, defining a set of project risks, or specifying elements of a project vision.

To model the actors for a context-level use case (context diagram), for example, you could ask, "Who or what provides information to or from the system?" or "Who or what must interact with the system?" When you need to order elements in a model, your question can direct participants to a sequencing task -- for example, "In what order will a user do these steps?"

Focus questions for requirements models can also build on some existing modeling element or project information. For example, if you have a list of actors and want to create a list of use cases, then you can ask, "What goals does the actor have when he interacts with the system?" Or, if you have defined use-case steps and now want to define exceptions, you can ask, "What can go wrong during this step?" This same question is also useful to elicit business rules, because exceptions are business rule violations.

Tools and Equipment for WoW Sessions

I prefer to use low-tech space for a WoW session because it's easy to change, it's dynamic, and frankly it's fun. Keeping your room visually rich and full of tools and materials invites creativity and change.

I equip the space with low-tech materials such as whiteboards, posters, butcher paper, sticky notes or cards, as well as tools such as colored markers. For the surface, I like to use a "sticky wall," which I create by covering two or three walls with poster roll paper that I've sprayed with repositional or remountable spray (available from 3M and other companies -- best to spray these rolls out of doors or where there's plenty of ventilation, then bring them inside to the WoW room). This makes a tacky surface to hold the cards (5X8 colored index cards work well) and lets you reposition them as you like. Alternatively, I use 6x8 sticky notes (repositionable sticky sheets sold by Vis-It or 3M) on a flat wall. An easy way to document the wall's content is to take a digital photo.

As a rule, I don't like to use complex software during the session because it can reinforce tendencies to overanalyze and introduce unnecessary model elements. Projecting word processor content on the wall, on the other hand, can be useful -- especially for text-based deliverables such as use cases, business rules, lists of issues, decisions, and next steps. A person designated as the recorder can enter such text items in real time and project them on a laptop projector, or at least print and distribute them. This is especially efficient during walkthroughs, when the group reviews its work to detect defects.

In one recent workshop, we used both a laptop projector and a printer to great effect. In subgroups, business participants hand-wrote their requirements (in this case, data requirements, business rules, and business justifications) onto a paper template. Then, the documenter entered these into a laptop while the group took a break. The requirements generated by everyone were printed and projected onto the wall for a walkthrough. Instead of waiting to receive corrected documents after the session, the participants made corrections in real time and walked out with a set of requirements.

High-tech tools are useful in a WoW session for both documentation and requirements tracing; the key is to not make the tool the focus of attention. In one workshop, we had an analyst working apart from the group to capture atomic business rules in a requirements management tool. Team members were too busy working with cards on the wall -- building and testing their business rules -- to worry about what the analyst was typing. He printed the rules from the repository in easy-to-read mini-reports several times a day, and the team used these reports, along with previously elicited scenarios, to test each other's business rules in subgroups.

In another workshop, a data analyst observed and recorded the group's data model in his visual modeling tool. This was useful because occasionally the tool triggered awareness that certain elements, such as cardinality and relationship rules, were missing from the wall models. The analyst made a list of these elements, and from this I created a list of quality assurance questions for subgroups to answer.

Nothing beats the speed and throughput of a same time/same place workshop that exploits the WoW approach. But a real-time WoW session also helps set the stage for continued collaboration when the team disperses. That's when automated tools really shine -- by keeping everyone informed about ongoing changes, by linking your requirements to other project elements such as architectural and design models, code, and tests, and by tracing your requirements forward through the development process. In addition, group members, -- who will have established a mutual trust during the workshop, -- can use collaborative software tools for same time/ different place sessions.

Engineering Collaboration into Your Work

Even with all our development tools and technologies, the process of building software still depends largely on the collaborative brainpower of the people who work on the project. The WoW is an effective technique for harnessing that brainpower. It fosters contributions from all individuals and provides a common place for creating emergent deliverables. By employing this simple, easy-to-use, repeatable way to build deliverables, you can accelerate your team's ability to collaborate, and begin engineering collaboration into the software development process.

Acknowledgements

I thank Daryl Kulak, Hong Li, Charlie Poole, Richard Specter, and Karl

Wiegers for their helpful comments and suggestions for this article.

Notes

1 Users, or end users, are the people who will interact directly with the software. Sometimes workshops use surrogates if end users are unavailable, but I make sure they have the content knowledge to truly represent user needs. Indirect users -- people who may not directly use the software - also may have relevant information about requirements (often business rules). Customers pay for the software. It helps if they are present to make decisions about scope and priority as well as trade-offs between functionality and other concerns such as time, cost, or nonfunctional requirements (reliability, speed, and so on). Of course, the people who design and create the software -- including developers, analysts, software engineers, testers, and project managers -- are key participants.

2 B. W. Tuckman, "Developmental Sequence in Small Groups." *The Psychological Bulletin*, 1976, No. 63, 384-399.

3 See www.icaworld.org.

4 For a complete explanation, see www.ebgconsulting.com.

5 See "Decide How to Decide: A Collaboration Pattern" under Facilitated Workshops at www.ebgconsulting.com.

6 Stakeholder classes (also known as user classes) identify the people who care about -- or should care about -- the systems being developed. Stakeholders include customers, users, and software suppliers, as well as people who have knowledge about the requirements. Stakeholder classes are particularly useful when you're looking to answer questions such as "Who will the system affect?" and "Who will affect the system?".

7 A context diagram (also known as a scope model, system context use case, or context-level use case) shows the system as a whole in its environment. The system is a bubble or oval in the center, external interfaces and actors are boxes and sticks figures, and inflows/outflows are arrows labeled with nouns. A context diagram is useful when you're looking to answer questions such as "What does the system get?" and "When do things happen, and what are the results?" and "What do external actors provide to and receive from the system?".

8 An event table (also known as an event list) defines the "triggers" for events to which the system responses. Some of these responses involve visible outputs; some involve only changes to internal information; some involve both. The table, which lists both the event and the response, is helpful for answering questions such as "When will things happen?"

9 See "Decide How to Decide: A Collaboration Pattern" under Facilitated Workshops at www.ebgconsulting.com.

References

Scott Ambler, "The Practices of Agile Modeling," 2001. See "Display Models Publicly" at http://www.agilemodeling.com/practices.htm.

Alistair Cockburn, "Software Development as a Cooperative Game," 1999. Available at http://members.aol.com/humansandt/papers/asgame/asgame.htm.

Ellen Gottesdiener, *Requirements by Collaboration*. Addison-Wesley, forthcoming in 2002.

Institute of Cultural Affairs, 1994. *Group Facilitation Methods* course guide, Institute of Cultural Affairs.

Jon R. Katzenbach and Douglas K. Smith, *The Discipline of Teams: A Mindbook-Workbook for Delivering Small Group Performance*. John Wiley and Sons, 2001.

Michael D. Schrage, *Serious Play: How the World's Best Companies Simulate to Innovate*. Harvard Business School Press, 1999.

Laura Spencer, *Winning Through Participation: Meeting the Challenge of Corporate Change with the Technology of Participation*. Kendall/Hunt Publishing Co., 1989.

Richard A. Zahniser, "Design by Walking Around." *Communications of the ACM*, October 1993, pp. 115-123.

For more information on collaborative workshops to create requirements deliverables, also see The Rational Unified Process, "Work Guidelines for a Requirements Workshop" and "Work Guidelines for a Use Case Workshop."

For more information on the products or services discussed in this article, please click here and follow the instructions provided. Thank you!