

Ellen Gottesdiener

Ellen Gottesdiener, Principal of EBG Consulting, Inc., is a consultant, facilitator and trainer helping people to collaboratively capture and verify business, user and technical requirements. Ellen is an expert in using facilitated workshops for software development projects. She is a pioneer in business rule-driven requirements workshops. As an independent consultant, she works with project teams to help them explore requirements, shape their development processes and collaboratively plan their work. Ellen presents seminars on requirements, facilitated workshops and peer reviews. She is a frequent speaker at numerous industry conferences and has authored numerous papers on requirements, workshops, methods and modeling. Her book *Requirements by Collaboration* is forthcoming from Addison Wesley. Ellen can be reached at ellen@ebgconsulting.com or through her web site, <http://www.ebgconsulting.com>.

Requirements Workshops: Collaborating to Explore User Requirements

Development VP Avery Kerr of BeWell Medical Corp. assigned our group to define the requirements for a software system to be used by registration agents and customers.

“They’re all over the place with what they want,” lamented Mary, RegTrak’s project manager. I asked her to explain what she meant. Who was all over the place? What did they want?

“Product development wants our global agents to use RegTrak to register and order products for their regions or countries,” she explained. “Distribution wants hospital purchasing agents to be able to place orders and check on them. Marketing wants health care workers in the hospital to have access, too.”

It was clear that the RegTrak team needed to nail down the scope and high-level requirements of the proposed project. Because Mary was newly assigned to RegTrak, she was especially concerned about getting the project off to a good start.

Her comment confirmed my typical experience: Each of the various stakeholders has his or her own understanding of what the scope should be, based on personal experience and knowledge.

“I suggest we bring together all of the stakeholders to define the scope of the project,” I said. Mary agreed. We formed a workshop planning team and got to work.

Surfacing User Requirements

We often talk about how tough it is to capture user requirements. The term *capture* implies that requirements are imprisoned and taken over. In reality, user requirements need to be acquired and kept in trust by the software organization. The ultimate knowledge of user requirements, and the decision-making with regard to the quality of those requirements, rests with the business people who sponsor a software project. At the same time, though, business people need to understand the impact that software will have on operations and prepare for change.

User requirements have always been difficult to surface and clarify. Many industry experts assert that defining user requirements is the most difficult task of software development. One of the biggest reasons is the communication gap between software and business people, which result in less than stellar customer-supplier relationships. Each side speaks using its own acronyms and shortcuts. Unless they’ve experienced each other’s roles, they have trouble truly appreciating each other’s legitimate concerns. This problem exists for both external software development organizations and software groups supporting internal business organizations. User requirements

models help bridge these communication gaps.

User Requirements Models

User requirements are the needs that users have (people or things that affect, or are affected by the system). User requirements can be expressed as functional hierarchies using text; for many business systems, models can be used to define user requirements.

A **requirements model** is a high-level blueprint for a software product. It takes the form of a diagram, list, or table, often supplemented with text, that depicts a user's needs from a particular point of view. Examples of user requirements models include event lists, use cases, data models, class models, business rules, actor maps, prototypes, and user interface navigation diagrams.

Whatever its form, though, the primary purpose of a requirements model is to *communicate*. Defining requirements is a discovery process for users and customers. User requirements evolve from the process of users “trying out” their requirements through models.

Requirements models speak to both users and software people. As users create and modify the models, their understanding of their requirements changes and evolves—in other words, the requirements become clearer. Defining requirements with models works independently of software development methodology or technology. If you're using a more formal, disciplined development methodology, you're likely to need more models that contain greater detail. With less formal and prescribed methodologies, you'll create just a few models, and the models aren't likely to be very precise. In either case, models are needed. This is true if you're using more traditional technologies and languages or bleeding-edge tools.

An effective way to develop user requirements models is to use requirements workshops.

Requirements Workshops

A **requirements workshop** is a structured meeting in which a carefully selected group of stakeholders and content experts work together to define, create, refine, and reach closure on

deliverables that represent user requirements. The workshop process nurtures team communication, decision-making, and mutual understanding. Requirements workshops are an effective way to bring together customers, users, and software suppliers to improve the quality of software products without sacrificing time to delivery. They tend to commit users to the requirements definition process and promote their ownership for the deliverables and, ultimately, the system.

Requirements workshops can bridge communication gaps among project stakeholders. Co-creating models in a requirements workshop expedites mutual learning and understanding. By asking focused questions in the workshop, the workshop facilitator helps participants define requirements at different levels of specificity. You might start with a scope workshop to outline the terrain, defining what the software will deliver and getting agreement on a starting set of functionality. If there's agreement on scope, you might begin with user interactions and evolve your detailed user requirements models from that point. In any case, workshops provide a forum in which customers and users can make informed decisions about delivery trade-offs and priorities.

A successful workshop requires the participation of key project stakeholders. Each workshop is treated like a mini-project that's woven into the fabric of a software or business project. Like any project, each workshop requires planning, role clarification, and infrastructure. It has a beginning, a middle, and an end. Deliverables are defined beforehand, but they'll change as the group learns.

Requirements workshops are based on the premise that a small group of knowledgeable, motivated people is more effective than one or two development "heroes." They're also based on the premise that, as Jerry Weinberg said, "One of us is not as smart as all of us." You tap the collective wisdom of the group to get your user requirements. This requires collaboration and facilitation.

Workshops: Collaboration and Facilitation

Collaboration occurs when all members of a group or team share a common purpose, there's mutual trust, and everyone uses agreed-upon approaches for the work. The members operate like a

jazz ensemble: multiple voices interwoven, playing together and individually, generously and inventively, sharing a single theme.

The most successful workshops are *collaborative*. This means that the participants share a common goal and join together to create **deliverables** that contribute to that goal. Collaboration doesn't just happen, though; teams don't just form and jell automatically. Collaboration needs to be engineered into a team's work. Requirements workshop participants should include a healthy mix of business and software people who share a common goal and have stakes in the same project. A key element of successful and collaborative workshops is that the participants agree to the workshop purpose, principles for participation, and products ahead of time. They also determine a decision-making process ahead of time; this permits the group to reach closure on their deliverables and the actions they will take after the workshop.

“Facilitation is the art of leading people through processes toward agreed-upon objectives in a manner that encourages participation, ownership, and productivity from all involved.”

— David Sibbet, *Effective Facilitation*, 1994, *The Grove Consultants International*, San Francisco CA.

A requirements workshop is a *facilitated* group meeting. Each workshop requires the design of a facilitated group process for the specific group needing to deliver requirements for a specific project. The interactions in the meeting enable requirements to be discovered, elaborated, clarified, and closed.

Roget's Thesaurus says that the *noun* “facilitate” means *easing, smoothing, expediting*, while the *verb* “facilitate” means *ease, grease the wheels, smooth, pave the way*. It's pretty likely that at least half of the meetings you attend in your work are unproductive (see sidebar “Meeting Madness”) and a waste of your time. A facilitated meeting is different from these meetings in several key ways that I explore in the next section.

Workshops Are Different from Typical Meetings

The requirements workshops described here are a modern-day variant of Joint Application Design or Development (JAD). On the surface, facilitated workshops are like meetings in that both

involve people meeting together at the same time, and both (presumably) follow a logical flow. But there are significant differences.

Some of the differences have to do with responsibilities:

- Meetings are led by a manager or leader, who generally has done minimal preparation; requirements workshops are led by a neutral facilitator who's prepared intensively.
- Meetings require little or no pre-work of the attendees; requirements workshops generally require a fair amount of pre-work, including the creation of products that serve as candidates for workshop inputs.
- The authority to make decisions in meetings rests with the meeting leader, and that authority may not even be a topic of discussion; in a requirements workshop, each decision has an associated decision rule, and there's a decision process, and authority can rest with one or more people (but not the facilitator).

Other differences involve format:

- Meetings are about information exchange; requirements workshops are about information discovery and creation.
- There's often not much interaction among people in a meeting; interactions among people in requirements workshops are intense and varied, and the participants perform activities as individuals, members of subgroups, and members of the group as a whole.
- Meetings allow little or no time for playful activities; requirements workshops encourage "serious play" in order to promote innovation and foster teamwork.

Still other differences are connected with the resulting documentation:

- Meetings rarely involve producing deliverables; requirements workshops involve participants creating and verifying deliverables such as requirements models.
- Meetings rarely require inputs in the form of work products; requirements workshops almost always require that participants create things like draft models.
- Meetings use visual media sparingly, if at all; requirements workshops make heavy use of visual media such as posters, sticky notes, cards, and diagrams.

Perhaps the most important distinctions between workshops and meetings are the presence of specific process roles—facilitator and recorder—in workshops and the fact that workshops deliver specific, predefined products.

A neutral **facilitator** assists the group by leading the workshop planning process and then guiding the group through the workshop. A person serving in another process role, the **recorder** (or **scribe**), might also assist by capturing the group's work in real time.

Neither facilitator nor recorder operates as a content expert, nor does either collaborate in product creation. As a result, they're free to focus on the process. As the group becomes more familiar with the workshop process, the facilitator's role in controlling the process can be relaxed.

The facilitator's job is to balance the needs of content, process, and people.

- Balancing *content* involves ensuring that the necessary user requirements are delivered at the right level of detail and with the necessary degree of quality.
- *Process* balancing requires the facilitator to design a sequence of activities that follows a logical progression within the specified time constraints, to promote participation, and to keep participants energized and engaged.
- Balancing *people* needs is also a key responsibility for the facilitator. This involves helping participants build their relationships, exploiting the strengths of different styles of thinking, learning, and interacting and helping participants become a high-performing group.

Workshops Deliver Products

A key difference between workshops and meetings is that workshops deliver specific, predefined products. These include tangible products, which are the requirements models. Workshops also deliver important intangible products, such as mutual learning, increased understanding, decisions, motivation, and teamwork. These products are specifically planned ahead of time.

During your workshop, participants use one model to test another, and in so doing uncover missing and erroneous requirements. Because each requirements model connects in some way to another, you can also use one model to elicit another. Participants can more easily see connections, and they quickly learn to iterate across multiple models in the workshop, which leads to more complete requirements.

The following are real examples of the work products created in facilitated workshops (Gottesdiener, 1999):

- In a half-day midpoint debrief (retrospective), a beleaguered project team created an action plan for correcting critical project problems, re-established how and when team communications will occur and redefined several key roles, including that of the project sponsor.
- In two hours, workshop participants generated 119 business events, classified them and then removed those not in scope during their requirements scoping workshop.
- In a use-case modeling workshop, business participants were able to test their use case model and structural business rules using 55 scenarios in less than 90 minutes.
- In three hours, a team validated a medium-complexity data model and made decisions about the scope of the data about which business performance metrics would be based.
- In a four hours, a project team defined the deliverables necessary for the upcoming design phase, who owned, reviewed and created each deliverable, what their quality criterion would be, and mapped them into dependencies along a timeline.

Workshop planners need time to define what to deliver, what level of detail the materials should contain, and how the intangibles will be achieved. A useful guideline is that you'll need at least one day of planning time to facilitate one workshop day. This is not insubstantial, but the benefits can be vast.

Other Uses for Workshops

Elements of requirements workshops, such as process planning, product delivery, and process roles, can be applied in other settings. For one thing, you can use facilitated workshops to launch a project, in order to deliver elements documented in the **project charter** or vision document. In fact, you might usefully create the scope-level requirements during the chartering process. Other good uses of workshops: strategic planning, process improvement, and problem solving.

Types of Requirements Workshops

You can use workshops to elicit, specify, review, quality-check, and reach closure on user requirements deliverables. Figure 1 shows a sequence of three types of requirements workshops. (Note that the deliverables are examples; they are numerous deliverables you can use in requirements workshops).

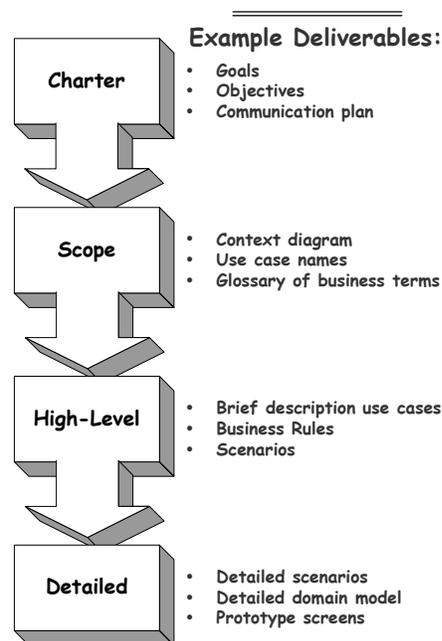


Figure 1: *Types of Requirements Workshops*

Although each project is unique, this diagram gives you a basis for defining the deliverables of any of the workshop types. For example, you might use multiple half-day workshops to deliver high-level requirements.

These workshops are preceded by an agreed-upon charter document or vision document and possibly a charter workshop. Typically, in a charter workshop, you'd deliver goals, objectives, a communication plan, roles and responsibilities, and perhaps some of the same deliverables of a scope workshop such as events, a context diagram and stakeholder classes.

You successfully refine user requirements, moving from a high level of abstraction to more fully intricate and detailed requirements. In a **scope workshop**, you take a birds-eye view of what user requirements should be included. The deliverables from the scope workshop clarify the items that need to be elaborated upon in your **high-level workshop**, which are in turn specified in the **detailed workshop**. You may combine workshop levels if your scope is fairly well-contained.

In practice, workshop categories can overlap. As you move through the workshops, you prioritize your requirements. In one of my workshops, we integrated prioritization with high-level modeling and then detailed modeling. I asked participants to write brief use case descriptions and then prioritize the whole set of use cases. From there, the participants created detailed use cases, scenarios, and business rules. In our last workshop activity, the participants revisited these priorities and adjusted the earlier list based on their deeper understanding.

Requirements workshops are often scheduled multiple times. Each workshop delivers a set of requirements, each of which is in a certain state or degree of completeness. You decide which products you need and tailor the outcomes appropriately. If time is of the essence, you might decide to deliver less precise user requirements, realizing that the trade-off will be more defects and rework during development. For projects in which quality is king, you'll want to deliver a more comprehensive set of requirements.

Making the Business Case

Some organizations shy away from workshops, shunning the time it takes to plan and execute them. Others claim that the cost of gathering people together in the same place at the same time is cost-prohibitive. Yet, requirements workshops can provide a substantial return on investment, for

example, \$10 return on every \$1 you invest (Jones, 1996).

Requirements workshops provide business value by reducing the time it takes to gather requirements, increasing team productivity, and reducing risks associated with software projects.

Aside from the hard data, requirements workshops contribute to customer and user satisfaction because they allow the people with the needs to participate directly in defining those needs. On one project, in which we built models like user interface prototypes during workshops in order to define requirements, not one high- or medium-priority defect originated with requirements.

The long-term prognosis for a project that uses requirements workshops is better because the workshops help build a healthy project community early in the project's life. In addition to delivering requirements, workshops also facilitate the forming of healthy team interactions, trust, and collaboration. They show by example the value of maintaining active customer involvement in the software development process.

When *Not* to Use Requirements Workshops

You need to use requirements workshops under appropriate project conditions. If your project is small and low-risk, I don't recommend using workshops. However, if your requirements are complex, and you value the side effects of collaboration—trust, mutual understanding, and participation—then requirements workshops may be for you.

Requirements workshops are most appropriate for business software or commercial software when you can get knowledgeable customers, users, or surrogates. Requirements for real-time systems are less visible to customers, so requirements workshops are probably not suitable.

Size matters, so be careful not to overload the workshops with too many people. The number of participants should not exceed 16 participants, unless you're going to use two facilitators. Most workshops have between seven and 12 participants. Each person you add to the workshop can add more time or reduce the number of deliverables. Strive for the minimum number of participants who can represent the collective needs of customers, users, testers, designers, and

analysts.

If you can't identify a workshop sponsor, you won't have the necessary commitment to ensure that the right players participate and prepare for a successful session. It isn't feasible for some organizations to gather users or their surrogates at the same time and the same place. In that case, you might consider using collaborative software tools that allow people to gather in a different time–different place virtual environment. Be aware that you'll need a skilled facilitator who can lead the process and also use the tool.

Requirements workshops require time to plan. If management won't allow time for planning and pre-work, your workshop results are likely to be mediocre.

Finally, don't use workshops unless you have a facilitator who is neutral to the outcome, experienced with group process, and knowledgeable about the deliverables you need to create in the session. These skills can be developed within your organization and shared across projects.

In lieu of workshops, you can use techniques such as observation, interviews, surveys, prototypes, competitive analysis, and product complaint data. Of course, these techniques can be powerful in combination with workshops, if workshops seem to be a fit for you.

Summing Up

There's no standard formula for requirements workshops. Each project, business situation, and group of people will combine to make each workshop unique. Preparing for the requirements workshop requires collaboration. It permits you to tap into the collective wisdom of all of the project stakeholders. In your workshops, participants are active, engaged, committed and task-oriented. A well-run workshop builds trust and mutual understand among all the participants. Workshops are not new, but are proven best practices in software development. They can go a long way not only in product delivery, but also in building a "jelled" team.

References

August, Judy H., 1991. *Joint Application Design: The Group Session Approach to System Design*, Yourdon Press, Prentice Hall.

Doyle, Michael and Strauss, David, 1976. *How to Make Meetings Work*, New York: The Berkley Publishing Group. Gottesdiener, Ellen, (spring, 2002). *Requirements by Collaboration: Workshops for Exploring User Needs*, Addison Wesley.

Gottesdiener, Ellen, "Specifying Requirements with a Wall of Wonder", November 2001, Rational Edge.

Gottesdiener, Ellen, "Collaborate for Quality: Using Collaborative Workshops to Determine Requirements", *Software Testing and Quality Engineering*, March/April 2001, Vol. 3. No. 2.

Gottesdiener, Ellen, "Decode Business Needs Now", *Software Development*, December 1999, Vol. 7. No. 12.

Jones, Capers, 1996. *Patterns of Software Systems Failure and Success*, Thomson Computer Press. Wood, Jane, and Denise Silver, 1995. *Joint Application Development, Second Edition*. John Wiley & Sons.

About the Author

Ellen Gottesdiener, Principal of EBG Consulting, Inc., is a consultant, facilitator and trainer helping people to collaboratively capture and verify business, user and technical requirements. Ellen is an expert in using facilitated workshops for software development projects. She is a pioneer in business rule-driven requirements workshops. As an independent consultant, she works with project teams to help them explore requirements, shape their development processes and collaboratively plan their work. Ellen presents seminars on requirements, facilitated workshops and peer reviews. She is a frequent speaker at numerous industry conferences and has authored numerous papers on requirements, workshops, methods and modeling. Her book *Requirements by Collaboration* is forthcoming from Addison Wesley. Ellen can be reached at ellen@ebgconsulting.com or through her web site, <http://www.ebgconsulting.com>