

Multi-modeling: Exploring Requirements to Define User Needs

Copyright by Ellen Gottesdiener, 2007.

This article first appeared in SearchSoftwareQuality.com, February, 2007.

To deliver the right product, you need to articulate your requirements early on, before the cost of fixing requirements errors kills your development budget, generates customer ill will and jeopardizes your business. If you don't obtain user requirements through efficient means, you won't deliver the right product.

The problem is that requirements are not just sitting around waiting to be written down. It takes a lot of effort and skill to elicit, analyze and verify them.

The most popular, traditional way to understand user needs is to write a set of textual requirements ("the system shall..."). But writing these statements is not enough. Textual requirements have their place when you need formal specifications, but they rarely communicate user needs effectively.

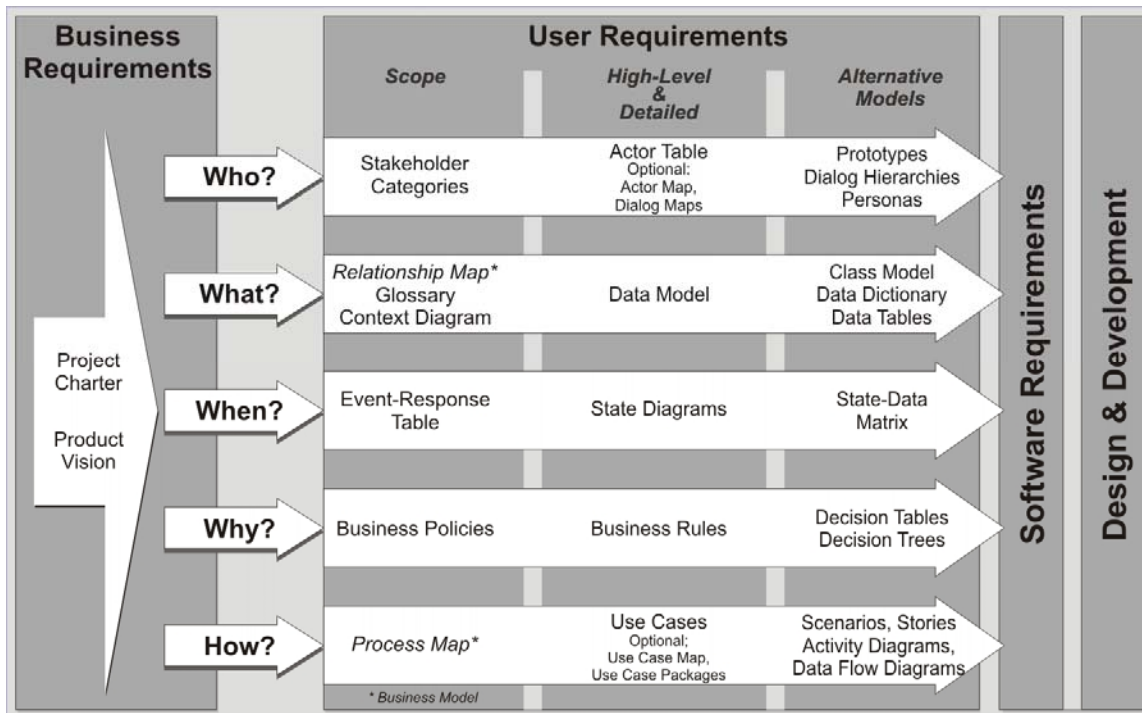
Understanding user needs is both an art and a science -- a combination of discovery, investigation and decision making. Successful projects engage users early and then explore and reach closure on their requirements by using *analysis models* – representations of user requirements.

What models tell you about your project

Models are approximations of reality. For software projects, analysis models depict user needs represented by text (such as tables, lists or matrices), diagrams or a combination of the two types. Model types include context diagrams, scenarios, data models, event-response tables, state diagrams and business rules.

Figure 1 shows how user requirements models answer the questions *Who? What? When? Why? and How?* Each model represents information at different levels of abstraction. For example, a state diagram represents information at a high level of abstraction, whereas detailed textual requirements represent a low level of abstraction. By first understanding the forest (a state diagram) before examining the trees (textual requirements), you can discover requirements defects that aren't evident when you review textual requirements alone.

Figure 1: Requirements Roadmap < Figure 1 ReqtsRoadmap-Gottesdiener.jpeg >



Copyright EBG Consulting, 2007

In my experience, analysis models created in collaborative workshops involving business and technical stakeholders are one of the quickest ways to articulate requirements, reveal missing and conflicting requirements, and crystallize user needs.

For example, recently I worked with a project team on Project TES. Before I came on board, the business owner and technical staff were passing many documents back and forth but making little progress in developing the TES requirements. It wasn't until we gathered together and started modeling requirements that the scope of the project became clear.

We used a combination of models -- a context diagram and an event-response table -- to clarify project scope. **Figure 2** shows the context diagram for project TES.

Figure 2: Context Diagram for Project TES



Source: EBG Consulting

The team created the context diagram in less than an hour. Meanwhile, a recorder captured matching events and their responses in a spreadsheet that was projected for everyone to see.

This modeling activity quickly revealed system interfaces that had not been understood by the technical staff before. It also raised questions about the source of some data -- questions that only the business people could answer.

Once the team recognized the complexity of the project, it was much easier for them to discuss an approach for tackling requirements in more detail. They concluded that they needed to define the requirements in logical chunks using iterations. A bit later, the team gathered to successfully elaborate the requirements using additional models (use cases, actors, scenarios, business rules, a state diagram and a prototype).

Seeing the biggest picture

In addition to representing user requirements before software specification, requirements models are useful for understanding an entire business process before you define a software solution.

For example, consider a project I'll call Project EX. A global supply-chain division wanted to alter its business processes, which involved forecasting, allocating finished goods and invoicing.

The business program managers assumed that they knew their software requirements. But after they launched the project, it became clear that there were gaps in their understanding of the overall end-to-end supply-chain process.

Again, analysis models came to the rescue. We gathered all the business program managers in a facilitated workshop to create a series of interconnected models -- a relationship map, a process map, state diagrams and scenarios. Figure 3 shows a portion of one of the state diagrams.

Using the models to explore their needs enabled the business people to significantly rethink the overall business process and to better understand the dependencies inherent in the supply chain. Consequently, the business owners changed their thinking about how software would help them solve their business process needs.

Figure 3: State Diagram for a Supply Chain (A Partial Consigned Purchase Order)



Source: EBG Consulting

The more, the merrier

Because requirements models are related, developing one model leads to deriving another. For example, when developing use cases, you can develop related models:

- Actors initiate use cases.
- Scenarios exemplify instances of use cases.
- A use case acts upon data.
- A use case is governed by business rules.
- Events trigger use cases.

The models thread together, so you can use various routes to harvest one model from another. You can develop the models quickly while at the same time verifying their completeness and correctness.

Using multiple interwoven models taps into different modes of human thinking and lets you explore different aspects of user requirements from different perspectives. Some people think more precisely with words, whereas others are better able to grasp concepts quickly by studying diagrams. Using both types of representation leverages different thinking modes and permits project stakeholders to understand requirements from more than one angle.

The bottom line

Requirements elicitation and analysis is a process of learning and discovery. The goal is to sufficiently understand your requirements so that your customers can prioritize them and allocate them to software. Models can supplement -- and, in some cases, substitute -- requirements specified as textual statements. Because each representation relates to another, using multiple models tends to accelerate understanding and reveal errors in requirements. Analysis modeling promotes overall software quality, helping you define the right requirements so that you can deliver the right solution.

Ellen Gottesdiener, Principal Consultant, [EBG Consulting](http://www.ebgconsulting.com), helps teams to collaboratively explore requirements, shape their development processes, and plan and review their work. Her book [*Requirements by Collaboration: Workshops for Defining Needs*](#) (Addison-Wesley, 2002) describes how to use multiple models to elicit requirements in collaborative workshops. Ellen's most recent book, [*The Software Requirements Memory Jogger: A Pocket Guide to Help Software and Business Teams Develop and Manage Requirements*](#) (GOAL/QPC, 2005), describes essentials for requirements development and management. Both are available on Amazon.com and via other quality booksellers.

Ellen presents seminars, speaks at conferences, and serves as adviser for numerous industry events, and she has authored numerous articles. She is a Certified Professional Facilitator (CPF) and a member of IEEE, IAF, ACM, DAMA, and IIBA. Ellen serves on the Expert Board of the International Institute of Business Analysis (IIBA) Business Analysis Body of Knowledge™ (BABOK™). EBG Consulting is a charter Endorsed Education Provider and [contributes to the IIBA](#) in a variety of ways.

You can subscribe to *Success with Requirements*, EBG Consulting's free monthly eNewsletter: <http://www.ebgconsulting.com/newsletter.php>. When you sign up, you'll receive a free .pdf article by our Senior Associate, Mary Gorman, on an essential requirements modeling technique. We created *Success with Requirements* to help you get the right requirements so your projects start smart and deliver the right product at the right time.