

Why Aren't YOU Doing Requirements Right?

*"We know
what we
need"*

In practice, project team members mostly don't know what users or customers need. Requirements development takes exploration and learning. It's unrealistic to expect your team to understand requirements up front.

For one thing, users, product managers, customers, and other stakeholders don't really know all their needs at the beginning. Requirements naturally thrash and evolve. Indeed, it's wise to be suspicious of claims to the contrary. Remember, *almost half* of the requirements you specify never get implemented (Standish Group International, 2003b).

In many projects, the perception that requirements are known is mistaken. Most errors in delivered software (30% to 50%, depending on the study) originate from flawed requirements (Schwaber, 2006; Nelson et al., 1999; Leffingwell and Widrig, 1999; Lauesen and Vinter, 2001).

The top three risks that threaten successful e-projects all relate to requirements— constantly changing requirements, poor requirements specification, and customer involvement issues such as delayed approval, requirements thrashing, and poor communication (Rodrigues, 2001).

Don't rely solely on product and business managers for defining user needs. Unless they are former users themselves, they will not understand direct user needs without inquiry and exploration. And rarely do product and business managers have the subject matter expertise you need to represent the entire set of requirements.

Ask yourself: have you solicited the voices of all your stakeholders? Do you know who all your stakeholders are? Have you prioritized conflicting needs? Have you explored both technical constraints and possibilities? You may think you know what the needs are, but your list may be shortened by technical realities or lengthened by technology possibilities.

What you *think* you know can hurt your project more than what you *don't* know.

"We've got this covered. We're [pick one: outsourcing/ using agile development methods/ buying a software package]"

Outsourcing, agile development methods, COTS solutions—these are often great ideas, but they don't eliminate the need to develop excellent requirements. You still need to articulate requirements, adapting your requirements development practices to these scenarios.

The critical need for proper requirements development *increases* when you outsource your project. You need to communicate requirements with even more rigor when development staff are not physically co-located with customers and project managers. In addition, you will need top-notch business analysts (Schwaber, 2006).

If you're adopting agile practices, it doesn't mean you don't need requirements. In agile projects, iterations are driven by requirements. They don't go away—they're successively elaborated.

And if your product is large and complex, agile projects need to start with a requirements-driven product and release roadmap. From there, the team develops chunks of requirements—based on those roadmaps. Success with agile development means balancing suitable-quality requirements with speedy definition of needs.

Many organizations hope to accelerate delivery by seeking and installing software package solutions (commercial off-the-shelf software, or COTS). In that case, you still need to understand your requirements and the impact your project will have on your business process. Requirements should drive your choice as well as your implementation strategy.

*"My staff
already
knows what
good
practices are"*

Too many projects rely on written requirements, often viewed as the most important good practice. But written requirements are rife with ambiguity (unclear meanings). To top it off, project and product needs are rarely known up front.

In fact, writing textual requirements ("the system shall...") is not the best way to understand your users' needs. Textual requirements have their place when you need formal specifications, but most successful projects also adopt other techniques to explore business and user requirements.

Effective requirements development makes use of requirements models that are verified and validated continually and iteratively. Using good practices—such as requirements modeling, facilitated workshops, prototypes, scenario verification, and more—takes practice, coaching, and reinforcement.

Following sound requirements processes, actively involving users, documenting requirements appropriately, validating and verifying requirements, and managing requirements changes—all these skills and techniques are essential to successfully reduce the many risks associated with requirements errors (Leishman and Cook, 2002).

"We can't afford to get training or consulting"

Roughly *one-third* of your software project budget is consumed fixing requirements errors. That means you're spending about \$150,000 of your \$500,000 project fixing defects or errors that originate from your requirements (Schwaber, 2006; Nelson et Al., 1999; Leffingwell and Widrig, 1999; Weinberg, 1997).

The earlier you discover these errors—missing, wrong, conflicting, and ambiguous requirements—the cheaper it is to fix them. Finding and fixing a requirements error during the requirements phase might cost you \$25 to \$100. If you don't find it until the construction or testing phase, fixing that same error is going to cost you \$500 to \$1000 (20 to 40 times more). Left undetected, that requirements error *will cost you as much as 300 to 1,000 times more*. That \$25 cost becomes \$10,000! (Reifer, 2007)

For every dollar you invest in your staff learning good requirements practices that incorporate customer collaboration, you can get a *10:1 return on investment* (Jones, 1996a).

You cannot afford *not* to correct your requirements deficiencies.

Pay now—or pay more, later.

"It will take too much time to do things differently, to take time out for training, or get project help from the outside"

Yes, there will be a learning curve. This is a normal part of change and learning. But there are things you can do to accelerate the process. Schedule formal training to coincide as closely as possible to the project work. Provide real-time coaching to the team. Set up sponsorship contracts so that new practices and behaviors are reinforced in the organization—both top-down and bottom-up. Find out from your staff what they need from you to be successful with requirements, and provide it.

And remember, some requirements work is better than none. On complex projects, one study showed that investing even 10% in the effort before freezing requirements reduces cost overruns significantly (NASA Comptroller Office, reported in Hooks and Farry, 2001).

Many organizations are turning to external service providers, outsourcing their development efforts. And they are learning that highly skilled business analysts who can develop and manage requirements are essential to successful outsourcing (Henschen et al., 2007; Light, 2005).

"Users don't know what they want"

Users are not supposed to know what they want. Understanding user needs is both an art and a science—a combination of discovery, interrogation, exploration, and decision making.

Involving users in requirements development is widely recognized as one of the—if not THE—most important factor for project success. Yet business people as well as IT people continue to complain about their inability to work effectively together to define the right requirements.

Healthy collaboration with users is crucial—and it doesn't just happen. Both sides of the relationship—business and IT—are accountable to co-develop the right requirements in the most efficient and effective manner possible.

That's why great analysts employ an appropriate combination of requirements elicitation techniques. It's one of their most valued skills. These elicitation skills, married with artful choices in requirements models, go a long way toward active and productive user involvement.

Sponsors who pay for the development (product managers, marketing managers, or internal business managers) also need to be engaged. This doesn't take unlimited time and money. Not all requirements are created equal. User priorities need to be evaluated continually if the team is to make smart product development choices.

"Customers are too busy to participate in requirements work with us"

IT needs to employ techniques that make good use of business people's time and actively engage them in requirements work. At the same time, business people need to fully participate in defining their requirements. If you do it right, good practices for effective user involvement sell themselves.

Here are some ways to do it right. Represent user needs in ways that "sing" to users and customers. Use a variety of elicitation techniques. Verify and validate requirements as you proceed. And, importantly, conduct continual requirements retrospectives to get feedback that will allow you to adapt your requirements practices.

"Our users are distributed. We can't get everyone's requirements"

User requirements are the focal point of your product. When users are scattered, you still need to identify the various types of users, understand their needs, and determine how you might need to alter requirements based on location.

When your users are geographically distributed or there are vast numbers of them, you may have to rely on surrogate users or subject matter experts who can research user needs. Find a small sample of representative users from various locations who are important to product success. Then adapt your elicitation practices to make efficient use of these users in requirements development and verification.

For some products, it's best to combine surveys and other research methods with deeper representative user involvement.

Regardless of the approach you take, ignoring user needs is a recipe for disaster.

"We got the book (The Software Requirements Memory Jogger). We'll just follow that"

Reading a book helps. It gives you awareness and knowledge. Reading does not, however, enable you to apply skills without practice and reinforcement.

Many business and requirements analysts are not trained and skilled in the toolkit of requirements development and management practices they need to be successful (Schwaber, 2006).

Analysts with extensive experience are more successful than novices in analyzing and uncovering user needs. Expert analysts demonstrate the ability to select among elicitation techniques based on the situation and integrate multiple models to represent requirements (Hickey and Davis, 2003).

Gaining expertise in requirements saves time and effort, reducing your total cost of application ownership (Light, 2005).

Training and coaching accelerate the learning curve and will earn you savings in time and money.

References

- Henschen, Doug, David Stodder, Penny Crosman, Michael McClellan, Neal Mcwhorter, and David Patterson. 2007. "Seven Trends for 2007." *Intelligent Enterprise*, January. See <http://www.intelligententerprise.com/showArticle.jhtml?articleID=196603897>
- Hickey, Ann M., and Alan M. Davis. 2003. "Elicitation Technique Selection: How Do Experts Do It?" *Proceedings 11th International IEEE Requirements Engineering Conference*. September.
- Hooks, Ivy F., and Kristin A. Farry. 2001. *Customer-Centered Products: Creating Successful Products through Requirements Management*. Amacom.
- Jones, Capers. 1996a. *Patterns of Software Systems Failure and Success*. Thomson Computer Press.
- Lauesen, Soren, and Otto Vinter. 2001. "Preventing Requirement Defects: An Experiment and Process Improvement." *Requirements Engineering*, June 6: 37-60.
- Leffingwell, Dean. 2003. "Calculating the Return on Investment from More Effective Requirements Management." *IBM Developer Works*, December.
- Leishman, Theron R., and David Cook. 2002. "Requirements Risk Can Drown Software Projects." *Crosstalk: The Journal of Defense Software Engineering*, April.
- Light, Matt. 2005. "Agile Requirements Definition and Management Will Benefit Applications Development." Gartner RAS Core Research Note G00126310, Gartner, April 18.
- Nelson, Mike, James Clark, and Martha Ann Spurlock. 1999. "Curing the Software Requirements and Cost Estimating Blues." *The Defense Acquisition University Program Manager Magazine*, November–December.
- Reifer, Donald J. 2007. "Profiles of Level 5 CMMI Organizations." *Crosstalk: The Journal of Defense Software Engineering*, January.
- Rodrigues, Alexandre G. 2001. "Project Goals, Business Performance, and Risk." *Cutter Consortium e-Project Management Advisory Service Executive Update*, 2(7).
- Schwaber, Carey. 2006. "The Root of the Problem: Poor Requirements." IT View Research Document, Forrester Research, September.
- Standish Group International, Inc. 2003b. *Standish Group Study*. Reported by Jim Johnson, chairman, XP 2002 Conference.
- Weinberg, Gerald M. 1997. *Quality Software Management, Volume 4: Anticipating Change*. Dorset House.
-