jama build great products™

Requirements Management 101 for Semiconductor Suppliers

Four Fundamentals Everyone Should Know

Make no mistake, engineers understand the fundamental value of requirements management. The challenge in implementing requirements management comes from the modern development environment itself.

Case in point: MRD, PRD and Technical Specs hundreds of pages long, solution fragmentation, and the fact that semiconductor companies must coordinate cross-functional deliverables of die, firmware, software and ecosystems. Combine those challenges with rapid advancements in algorithmic reasoning, semiconductor memory and sensor technologies, and we've touched on just a few of the many complications companies face.

While semiconductor engineers "get" RM, not all stakeholders do. Executives might not care about CMMI, BABOK or the details of functional specifications, but they do care about delivering what was promised to customers on time—and that's the value of requirements management. As a starting point, everyone involved should have a basic understanding of what requirements are and how to manage them.

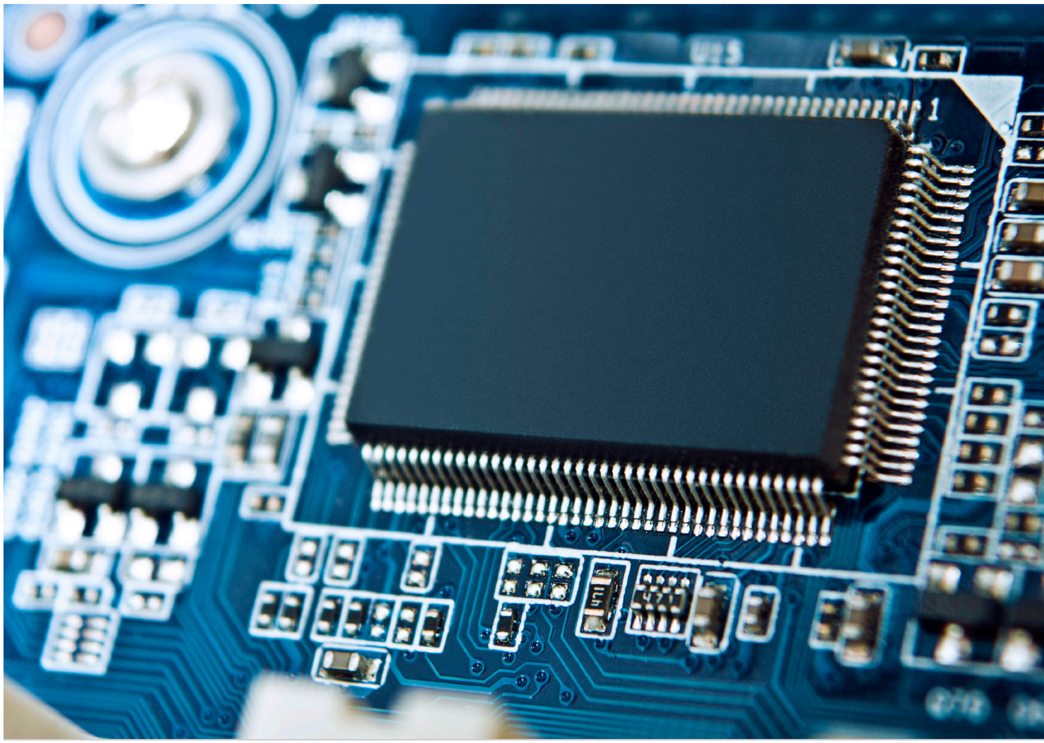
How Successful Teams Manage Requirements

Behind every delay or product failure are poorly managed requirements, and inadequate RM is usually the result of poor, unstructured communication between product planning and engineering. That's a big problem when changes occur, as the products themselves are increasingly complex.

A valuable semiconductor requirements management process keeps teams in-sync and provides stakeholders with project visibility. Success depends on everyone understanding and agreeing on what you are building and why—that's how we define requirements management. The "why" provides context for the goals, feedback and decisions about the requirements. This context increases predictability of future success and potential problems, allows your team to quickly course-correct, and greatly increases the odds of releasing an on-time and within-budget product.

Start with the Basics

A requirement is a statement that defines what you want to achieve or create; it identifies what a product needs to do and what it should look like, and it explains its functionality and value. A requirement can be expressed with text, sketches, detailed mockups or models—whatever information best communicates to an engineer what to design, and to the verification team what to verify.



Depending on your development process, you might use different terminology to capture requirements. High-level requirements are sometimes referred to simply as “needs” or “goals”. Within hardware development, the requirements might describe specific functionality and performance needs. Within software development, the requirements might describe functionality as “epics” and “stories.”

Regardless of what your team calls requirements or what process your team uses, requirements are essential to the development of all products. Without clearly defined requirements you could produce an incomplete or defective product. Throughout the process there can be many people involved in defining semiconductor requirements. A stakeholder might request a feature that describes how the product will provide value by solving a specific problem, or set of problems. An architect might create a system requirement that adheres to specific technical or organizational constraints.

> **It’s imperative that teams be able to access, collaborate, update, and test each requirement through to completion.**

Today’s sophisticated products require hundreds or thousands of requirements to sufficiently define the scope of a project or a release. Thus, it’s imperative that teams be able to access, collaborate, update, and test each requirement through to completion, as requirements change and evolve during development.

Now that we’ve defined the value of requirements management, let’s go deeper into the four fundamentals that every team member and stakeholder benefits from understanding:

1. **Planning good requirements**
   “What are we building?”
2. **Collaboration and buy-in**
   “Just approve the requirement, already!”
3. **Traceability and change management**
   “Wait, do the developers know that this changed?”
4. **Verification**
   “Hello, did anyone verify this thing?”

1 Planning Good Requirements

So what makes a good requirement? A good requirement should be valuable and actionable; it should define a need as well as provide a pathway to a solution. Everyone on the team should understand what it means. Requirements vary in complexity. They can be rough ideas sketched on a whiteboard to structured “shall” statements. They can be part of a group with high-level requirements broken down into sub-requirements. They may also include detailed parametric information describing how well the product should perform and how it interfaces with the system.

Good requirements need to be concise, specific and parametric and should answer the question, “What do we need?” rather than, “How do we fulfill a need?” Good requirements ensure that all stakeholders understand their part of the plan; if parts are unclear or misinterpreted the final product could be defective or fail. Good requirements also ensure that performance goals are clearly understood and agreed upon.

Preventing failure or misinterpretation of requirements can be aided by continuous feedback from the team as requirements evolve. Continuous collaboration and buy-in with everyone is the key to success.

2 Collaboration and Buy-In

Is everyone in the loop? Do we have enough requirements approval to move forward? These questions come up during development cycles. The truth is, for large projects with many stakeholders, getting complete consensus is rare. At best, it can cause decisions to be delayed, or worse, not made at all. Instead, get buy-in from the larger group and approval from those in control so you can move the project forward.

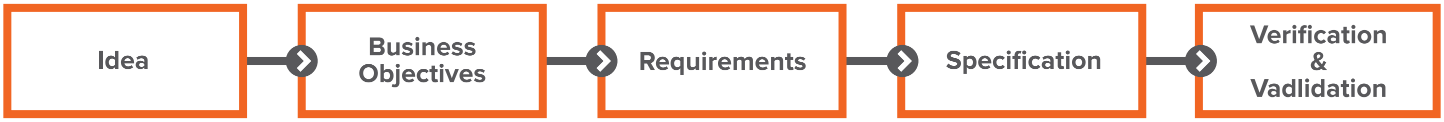
Consensus requires everyone to compromise and agree on the decision. Buy-in asks people to back the best solution and do what is necessary to move forward. You don’t need everyone to agree that the decision is perfect. You do need everyone to support the decision.

A structured team collaboration process helps everyone. Engineering receives direction on requirements planning negotiations and decisions, and everyone has a stake in the project and a venue for providing feedback. Collaborative teams continuously share ideas and tend to support decisions because they understand project goals. It’s when design engineers, verification engineers, test engineers or other stakeholders feel “out of the loop” that communication issues arise, people get frustrated and projects get delayed.

Once everyone has bought-in to the scope of work, requirements must be clarified and documented. Keeping track of all the requirements is where things get tricky. Imagine having a to-do list a mile long that involves collaborating with multiple people to complete it. How would you keep track of each item? How would you track how one change to an item affected the rest of the project? This is where traceability and change management add value.

3 Traceability & Change Management

Requirements traceability is a way to organize, document and keep track of each requirement from the initial idea through testing. Traceability identifies the relationships between items within your project. For example, a common downstream flow might look like this:



You should be able to trace each of your requirements back to its original business objective. By tracing requirements, you can identify the ripple effect changes have, see if a requirement has been completed and know if it's being tested properly. Traceability and change management provide managers peace of mind and the visibility needed to anticipate issues and ensure continuous quality.

Think of traceability as coverage that ensures your product meets all the vital requirements. Because requirements come from customers, engineers and many sources, each person contributes different requirements to the project. By tracing requirements, you ensure your entire team stays connected to the interdependencies of other items and the people working on those items.

Managing change is critical for preventing project and release "scope creep." Scope refers to "the work that needs to be accomplished to deliver a product with the specified features and functions." Scope creep refers to unplanned changes in development that occur when requirements are not clearly captured, understood and communicated. Good requirements provide a clear understanding of the breadth of work required to achieve the end product. This leads to a better development schedule and budget, which prevents delays and cost overruns.

4 Verification

Getting semiconductor requirements right results in better quality, faster development cycles and higher customer satisfaction with the end product. Requirements management also helps your team save money and prevent headaches throughout the development process. Concise, specific requirements can help you detect and fix problems early, rather than later when it's much more expensive to fix.

> **Research shows that project teams can eliminate 50-80 percent of product defects by effectively managing requirements. In addition, it can cost up to 100 times more to correct a defect after it's been coded than it is to correct as a requirement. By integrating requirements management into your quality assurance process, you can help your team increase efficiency and eliminate rework. And, rework is where most of the cost issues occur. According to the Carnegie Mellon Software Engineering Institute, "60-80 percent of the cost of software development is in rework."**

In other words, development teams waste the majority of their budgets on miscalculated efforts. For example, a developer codes a feature based on an old specification document, only to learn later that the requirements for that feature changed. The best protection and prevention is robust requirements management.

To learn more about requirements management and related development topics, browse through the white papers, case studies, videos and more at http://www.jamasoftware.com/resources/.

About Jama Software

Jama Software is the definitive system of record and action for product development. The company's modern requirements and test management solution helps enterprises accelerate development time, mitigate risk, slash complexity and verify regulatory compliance. More than 650 product-centric organizations, including NASA, Boeing and Caterpillar use Jama to modernize their process for bringing complex products to market. The company is headquartered in Portland, Oregon. For more information, visit www.jamasoftware.com.

Credits

Endnotes on articles referenced within Requirements Management 101.

1. Kupersmith, Jonathan. "Don't Bother Building Consensus." Business Analysis Times. 2011. Web. 2011. http://www.batimes.com/kupe-kupersmith/dont-bother-building-consensus. html

2. A Guide to the Project Management Body of Knowledge. 4th ed. Project Management Inst, 2008. Printed.

3. "Effective Requirements Definition and Management." Borland Software Corporation. 2009. Web. http://www.borland.com/resources/en/pdf/solutions/rdm_whitepaper.pdf

Other Articles Referenced and Recommended

"Business Requirements Analysis." MindTools. Web. http://www.mindtools.com/pages/article/newPPM_77.html

Frederick, Richard. "Introduction to Requirements - The Critical Details That Make or Break a Project." Global Knowledge. 2007. Web. http://www.globalknowledge.fr/PDF/WP_IntroRequirements.pdf

Larson, Elizabeth & Richard. Project Times. Web. http://www.projecttimes.com/articles/requirements-management-101.html

Ludwig Consulting Services, LLC. 2009. Web. http://www.jiludwig.com/

McMahon, Chris. "The Full Team Approach to Managing Requirements." Search Software Quality. Serena, Web.

"Real Reuse for Requirements." MKS. Web. http://www.mks.com/resources/data/documents/whitepapers/instances/real-reuse-for-requirements-14

Robin, Goldsmith F. "Four Tips for Gathering Requirements for the New Business Analyst." Search Software Quality. IBM, Web.

Wikipedia. Requirements Management, Traceability, Scope, 2011. Web. http://www.wikipedia.com