On Reqs. and Specs.: The Roles and Behaviors for Effective Product Definition

Steve Johnson & John Milburn
Pragmatic Marketing, Inc.
sjohnson@pragmaticmarketing.com
jmilburn@pragmaticmarketing.com

Abstract

There continues to be confusion between the product management role and the development role with respect to the form and level of technical specificity of requirements. This paper identifies the fundamental issues around market domain-focused, problem-oriented requirements definition, versus technical specification writing.

1. The Problem

Can you believe the angst between product managers and developers? Many product managers think their developers are incompetent. The feeling is mutual: developers don't think product managers bring any value to the process. Yet we all want to get products to market but our understanding of roles keeps getting in the way. Who writes requirements? Who writes specifications? And what's the difference between a req and a spec? The requirement is the problem.

Product managers write terrible requirements, littered with buzz words, ambivalent language, and non-specific performance parameters. They read like somewhat-technical marketing hype. And developers have to make sense of the requirements. They complain, "I cannot program to these requirements." And they're right.

So product managers try to become more specific by writing what we call ReqSpecs. Part problem, part implementation--and impossible to use. Developers complain "don't tell me how to do my job" because the requirement now explains how the feature should be implemented. And they're right again.

2. The Role of Product Management

The primary role of the product manager is to be the "messenger of the market", and we must not take on additional tasks until that primary one is completed. Yet product managers frequently attend design meetings. Why? What value do they add? If product managers have time for design meetings, it had better be after they have completed their on-site visits with their market segment.

Besides, product management attending design meetings is not about getting better designed products; it's about sharing blame for poor design. "What do you mean you don't like it? You were there when we designed it!?!?!" If product managers are creating the design, what value is development providing? Coding? If that's all, perhaps we should just fire all the developers and outsource to India or Ireland or someplace where at least they will code to our design. That is, we'll get exactly what we asked for without the developer's editorial license.

Developers say they cannot program to product management's requirements. And they're right. What they need is a functional specification. ReqSpecs don't solve the problem. Let's clarify some terms.

- A requirement is short statement of the problem

- A specification is how to solve the problem

3. Requirements vs. Specifications

In Pragmatic Marketing's seminar *Requirements That Work*™, we teach product managers how to log market problems, distill them into the unique set of problems, and write requirements based on the market problems. A Market Requirements Document (MRD) is a master list of all problems and the number of customer and prospect sites reporting the problem. Developers value the class as much as the product managers because eliminating dysfunction by teaching product management to deliver what developers so desperately want to know: what are the problems in the market? The problem is the requirement

By definition, a requirement is implementation-free; that is, absent design. When developers request more specific implementation details, they're asking for a specification instead of a requirement. In eXtreme Programming (XP), a requirement fits on an index card and is delivered in the form of a story. The requirement is also an explicit "agreement to discuss" so that developers fully understand the problem.

Instead of being in the requirement, implementation details must be in the specification. A specification is the *architect's intended implementation* to solve the problem. It is not an architectural blueprint of the final product.

Joel Spolsky has already written much on specs. We defy you to find an article that doesn't have at least one tip on creating better products. He writes:

> *"On any non-trivial project (more than about 1 week of coding or more than 1 programmer), if you don't have a spec, you will always spend more time and create lower quality code."*
> *Source:* www.joelonsoftware.com/articles/fog0000000036.html

> *"A functional specification describes how a product will work entirely from the user's perspective. It doesn't care how the thing is implemented. It talks about features. It specifies screens, menus, dialogs, and so on. A technical specification describes the internal implementation of the program. It talks about data structures, relational database models, choice of programming languages and tools, algorithms, etc."*
> *Source:* www.joelonsoftware.com/articles/fog0000000034.html

So a requirement states the problem. A specification states the solution. Joel defines two kinds of specification: functional, which is the intended approach to solving the problem, and technical, which is a detailed internal implementation. In the end, we all have a critical role to play:

- the product manager finds and quantifies market problems, articulating them in the form of requirements

- the product architect (or designer) writes a functional specification describing the approach to solving the problem

- the product developer creates a technical specification that fully describes how the functional specification will be implemented

Think developers are incompetent? Maybe they're working in a dysfunctional environment and are frustrated themselves. Focus on market problems and you're halfway to a reasonable solution. Design a reasonable solution to the problem and you'll deliver a product that people want to buy.

4. Well-Written Requirements

We've completely abandoned the traditional "the product shall" approach to writing requirements. Instead, we encourage product management to focus on the problems and let development focus on how to solve the problems. Many of the requirements that we read are in fact thinly- veiled feature descriptions.

Do you write requirements in the form of "the product shall?" If so, then you are probably limiting the creativity of your development team and writing specifications. However, depending on the requirements writing legacy that you inherited (probably from a extremely project- or program-oriented waterfall environment), you will need to transition your teams to understand that the product management role does not include writing specs, but rather that the value they add is through customer interaction and understanding market problems.

Do you write "as a 'role,' I want to 'perform and activity,' so that I can 'achieve a goal'?" If so, then you are probably an agile shop, using a variation of Scrum or XP development techniques. We have found this method seems to work the best in software companies, with co-located teams, that have a fairly high degree of domain expertise. However, there are numerous examples of multi-national projects building hardware or electrical components that have also been successful with this approach – it just requires different skills and techniques to be successful.

Our preference is the Requirements That Work format: [Persona] has [problem] with [frequency]. It forces product managers to explore the problem, not the solution, and helps the design team understand the context of the problem.

The benchmark for well-written requirements is:

- Is there a clear definition of the user(s)?

- Do I understand their problem / what they are trying to achieve?

- Do I have supporting documentation that provides the context about the persona and their problems so that I clearly understand how to design a solution to their problem?

5. Designer, Architect, or Product Owner?

If your company is considering introducing a new artifact, our guess is that you really need to introduce a new role! You're probably missing a designer. Product managers have a full time job indentifying and quantifying the market problems. Your developers probably have their hands full delivering solutions. But who is designing those solutions?

No hardware company would build products without a designer, why do software companies? A designer analyzes the problem and designs a solution. Their output is a specification that the developers will use to develop to.

Alan Cooper argues that most projects fail because they do not have a spec at all. This is because most companies do not have product designers or architects. Product managers create desired feature lists. Developers write code. But no one is designing anything. Would you hire a carpenter to build the house? Of course not. You would hire an architect! Yet programmers write complex software products without a design. So Cooper suggests that we add a role that is new to most development organizations: the **product architect**. The architect analyzes the problems that are described in the requirements and then creates a specification, at a functional level, how the problem can be solved with the product. The architect delivers a recommended approach to solving the problem, serving as the bridge between product management and development.

Pragmatic Marketing's 2008 Annual Product Management and Product Marketing Survey found that for each product manager, there is an average of 1 product architects/designers, 0.72 development leads, and 5.5 developers. (www.pragmaticmarketing.com/publications/survey/2008/2008-annual-product-management-and-marketing-survey). However, when these same product managers were asked how they spend their time, almost 50% are still writing detailed specifications! So, what are the architects and developers doing?

Product Management and Product Marketing Activities – 2008
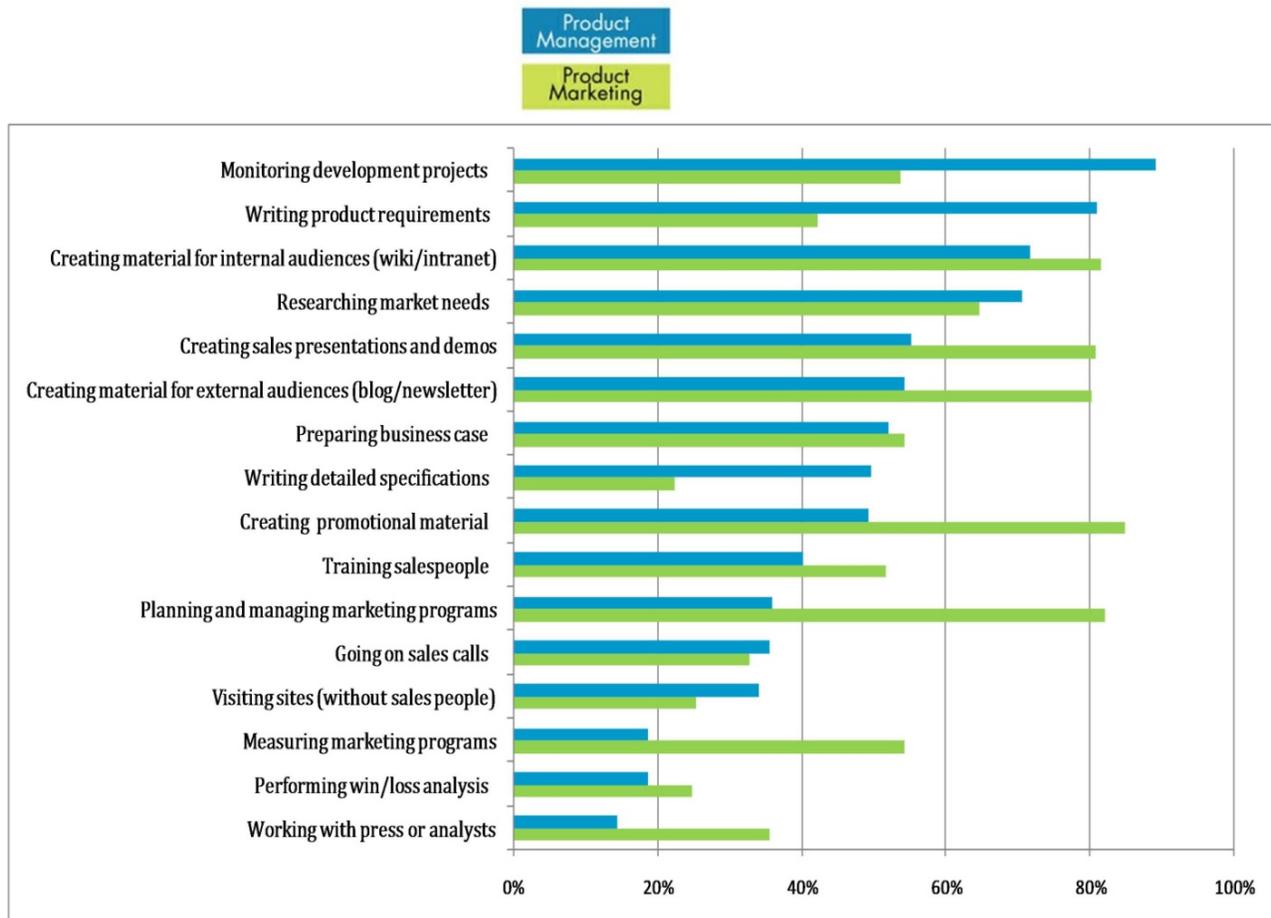
Regardless of the title or organizational structure, it is imperative that there is a clear delineation of the "what" from the "how" – development does the how, and product management does the what. Who's at fault? We have found that it is usually one of three fundamental reasons that this is not being done:

1. Product managers, especially those who came *from* a development role, write specifications rather than requirements because they are most familiar with them; don't know any better way to write them; and see this as a "comfort zone." Getting out and understanding the market takes time and is a lot harder than writing technical or functional specifications.

2. Regardless of the official title of the architect/designer role, it is critical that this individual(s) has the respect and understanding of the development team; has a close relationship with the product manager; and understands the customer domain as well as the technical domain. Lacking any one of these three can create gaps.

3. Some organizations do not fund the architect/designer role, and product managers act in this role by default. We see this especially in smaller companies who don't have the size or finances to fund the role.

Developers complain that product management requirements are not detailed enough. Requirements should in fact be absent design. Instead of providing more content, product managers should provide context so that the development team can fully address the problem.

Pragmatic Marketing Framework™

A Market-Driven Model for Managing and Marketing Technology Products

TACTICAL

STRATEGIC

	MARKET	STRATEGY	BUSINESS	PLANNING	PROGRAMS	READINESS	SUPPORT
							Presentations & Demos
						Sales Process	"Special" Calls
Marketing Plan				Launch Plan		Collateral	Event Support
Customer Acquisition	Positioning	Business Plan	Requirements	Thought Leadership		Sales Tools	Channel Support
Customer Retention	Buying Process	Pricing	Use Scenarios	Lead Generation		Channel Training	
Program Effectiveness	Buyer Personas	Buy, Build or Partner	Status Dashboard	Referrals & References			
	User Personas	Product Profitability					
		Innovation					
	Market Definition	Product Roadmap					
	Distribution Strategy	Product Portfolio					
	Product Portfolio	Competitive Landscape					
Market Problems	Win/Loss Analysis	Distinctive Competence	Technology Assessment				

PRAGMATIC MARKETING®

(480) 515 - 1411 ■ PragmaticMarketing.com