Writing the Market Requirements Document

By Steve Johnson
Pragmatic Marketing
April 2001

This document is written for product managers in high-tech companies who are chartered with documenting product requirements in the form of a Market Requirements Document (MRD).

Roles & Responsibilities

Writing a requirement is not a mysterious black art although it may sometimes seem that way. We write *requirements* to convey the details of a customer's *problem* to the reader (usually the developer). A requirement tells the reader the information necessary to innovate a solution to the problem, perhaps in ways not anticipated by the requirement's writer or associated end-user.

In order to understand how to write requirements, we must first understand the roles and procedures for developing products.

In many organizations, the product management function is ill defined. For the purposes of this document, a *product manager* finds problems in the marketplace and conveys these problems to the Development organization. Of course, this is in addition to other duties. These personnel are twice as likely to be in the Marketing department than in the Development department.[1]

Too often, product managers are overwhelmed with tactical activities that prevent them from representing the customer; we should ensure that activities that do not require a product manager are owned completely by Development. Many times, the product manager is competent in Development areas and is recruited to assist. Since many of us come from a technical background, we have a natural tendency to work on activities that are "fun" even though they are not product management.

Successful organizations follow these five classic steps of developing products:

- find a problem,
- analyze it,
- design an innovative solution,
- code to the design, and
- test the result

In the business world, these steps are not always followed. Too few product managers communicate lucid requirements. Too few developers analyze the problem and design an appropriate solution. Yet successful organizations consistently follow all five steps; these steps are required.

A **Product Manager** owns the first step, conveying the problem to Development in the form of a requirement. The remaining steps are owned by Development. Increasingly an **Architect** or **Program Manager** controls the middle steps (analyze and high-level design) with a **Development Manager** in charge of the detailed design, coding and testing. If there is no Program Manager, then analysis and design falls to the Development manager. **Quality Assurance** is responsible for internal testing. The QA department also validates that the specifications (written by Development) answer the problems articulated in the requirement (written by product management) and then tests the code to ensure that the requirement is met; this verification role is more effective than merely attempting to "break the code" in clever ways. So the functions (and roles) are:

- find a problem (product manager)
- analyze it (program manager)
- design a solution (program manager)
- code (developer)
- test (QA)

Definitions

We expect to see 'personas and goals' increasingly used as the primary method of defining complex projects, with an associated reduction in the traditional 'reqs and specs" approach. Using 'Personas and Goals' puts emphasis on the final result rather than on the tedious implementation of individual features. This approach requires more analysis and design by Development, making it more attractive to competent development organizations.

Buyer. There are three types of buyers: economic, technical and users[2]. Only the "user buyer" uses the product; yet the others are involved in the buying decision. The economic buyer is concerned with return-on-investment; the technical buyer determines how the product will "fit" the company's technology environment. The user buyer is one who actually uses the product in production (also called a persona). We will not write products for the buyer but for the user persona.

Persona. A *persona* is the "ideal" operator or user of the product. There are usually only one or two personas using the product frequently.[3] The persona defines *who* will use the solution.

Goals. The personas have goals that indicate what they hope to achieve. A *goal* articulates the end-point more than the detailed steps in-between, allowing Development to be more creative in how we achieve the goal. An understanding of goals determines the order of development deliverables.

Persona	Who
Goals	When
Requirement	What
Use-Case	Why
Specification	How

Problem. A business situation that causes difficulty or concern to an existing customer or prospective buyer is called a *problem*.

Requirement. A statement identifying a capability, physical characteristic, or quality factor that bounds a product or process need for which a solution will be pursued.[4] A requirement defines *what* needs to be done. A requirement can define a feature for either a new or an existing product.

Use-Case. A specific example of a goal or requirement in production is called a use-case. The use-case explains a common scenario to further illustrate *why* the requirement is necessary.

Specification. A specification summarizes the analysis by explaining *how* the requirement will be met. Prototypes are a special form of specification, showing a proposed solution based on analysis.

Design. A design documents the specific details of the specifications implementation. This document need not be published beyond the Development team.

Market Requirements Document. The Market Requirements Document (MRD) articulates the new product or new release plan including personas, buyers, goals, use-cases, requirements, and specification sizing.

Personas

Personas are the customers who use the product in production daily or frequently. As stated before, there are typically only one or two primary personas. A persona document includes a biography of a fictitious but realistic archetype of a customer.

Sample Persona: Michelle

Michelle is a high-school senior. She has a desktop computer with access to the Internet via dial-up to AOL. She is competent with the computer but not a power-user. While it is hers, her dad chose the computer and set everything up. She and her friends spend hours chatting on AOL. She checks email daily and uses Word weekly for school projects.

Michelle has hundreds of audio CDs and hundreds of MP3 files on her hard drive. Her computer has a CD-Recorder (CDR) that she thought would be cool but she can't really use it since the included CDR software is lousy.

Goals

The *persona* defines a "best" or ideal user; the *goal* describes an end-result that might be achieved in many ways. Requirements are more detailed but more confining on the developer; goals are more open-ended and allow (and require!) more creativity and innovation by the Development team. For example, a requirement might state that the results shall use "flour, sugar, eggs, milk" while a goal states "birthday cake." And a satisfying solution to the goal might well be an ice cream cake with no flour or eggs.

Sample Goal: Make An Audio CD

Michelle would like to create a mix CD of the one or two "best songs" from a series of discs and from her installed MP3s. She'd like to create a CD she can listen to in the car or in her stereo. She will not continue to make audio CDs if the process is tedious, as she fears it will be.

Problem

Stated in the context of a specific persona, a problem represents a situation that our technology should be able to address. Problems are specific situations that can be clearly expressed.

Sample Problem: Michelle must type track names

When copying CD files to CDR, the files are listed as track1, track2, etc. To make a CD jewel case insert, Michelle must type all track names manually.

Requirement

After discovering problems in the prospect or user community, the product manager writes specific **requirements** that *document the problem and a potential solution to the problem*. The Market Requirements Document combines the many requirements into a coherent whole. For a new product, a requirement states a business problem that the potential customer is having that will be addressed; for an existing product, the requirement might instead document a usage problem for existing customers.

A requirement should be written in the words of the personas, in business or personal terms, rather than offering a technical definition. A requirement should be short, only one or two paragraphs, never more than a page or two.

IEEE recognizes four types of requirements:

- **Functional.** Observable capabilities that must be present for the persona to complete their goals or perform the task specified by the use case.

- **Performance.** Characteristics of capacity, speed, concurrency
- **Constraints.** Conditions that legitimately limit the design.
- **Interface.** Defined interaction with hardware and software components.

The following elements are included in a requirement document.

Name. A unique "nickname" for a shorthand reference to the requirement.

Description. In 50 words or less, describe the requirement to the problem or business challenge.

Category. To simplify reporting later on, categorize the requirement in logical groups, such as architecture, user-interface, APIs, and reporting.

Persona. Which persona is affected by this problem?

Type. What type of requirement (use a one-letter code in brackets)

[F] Functional.

[P] Performance.

[C] Constraint.

[I] Interface.

Source. Where did this problem or idea originate? Ideally, requirements should have a source of a customer or prospect rather than an internal source. References to all associated input documents aid in determining the pervasiveness of the problem.

[PP] Prospect Problem (from a qualified non-customer)

[ER] Enhancement Request (from an existing customer)

[PP] Prospect Proposal (such as RFP, RFI, ITT)

[LA] Lost account (prospect or customer)

[RD] R&D

[EM] Executive management team

[SC] Sales Channel

[CS] Customer Support

[TD] Technical Documentation

[PM] Product Management

Tracking identifier, author & document date (original and revised).

These fields should be adequate for articulating and tracking requirements in your organization. A summary table of the requirements is included with the Market Requirements Document.

Sample Requirement: Get Track Info

Name: Get Track Info

Description: The product shall automatically capture track information about the songs that have been selected for recording.
Persona: Michelle

Category: Recording

Type: Functional

Source: PM

Characteristics of a Good Requirement[5]

Necessary. The stated requirement is an essential capability, physical characteristic, or quality factor of the product or process. If it is removed or deleted, a deficiency will exist, which cannot be fulfilled by other capabilities of the product or process.

Concise (*minimal*, *understandable*). The requirement statement includes only one requirement stating what must be done and only what must be done, stated simply and clearly. It is easy to read and understand.

Implementation free. The requirement states what is required, not how the requirement should be met. A requirement statement should not reflect a design or implementation nor should it describe an operation. However, the treatment of interface requirements is generally an exception.

Attainable *(achievable or feasible).* The stated requirement can be achieved by one or more developed system concepts at a definable cost. This implies that at least a high level conceptual design has been completed and cost tradeoff studies have been conducted.

Complete *(standalone).* The stated requirement is complete and does not need further amplification. The stated requirement will provide sufficient capability.

Consistent. The stated requirement does not contradict other requirements. It is not a duplicate of another requirement. The same term is used for the same item in all requirements.

Unambiguous. Each requirement must have one and only one interpretation. Language used in the statement must not leave a doubt in the reader's mind as to the intended descriptive or numeric value.

Verifiable. The stated requirement is not vague or general but is quantified in a manner that can be verified through inspection, analysis, demonstration or test.

Use-Case Scenarios

Two or three use-case scenarios explain to the developer why a high-level goal (or detailed requirement) is needed. The use-case illustrates how the problem occurs and how a solution might be used. The tone is "imagine if you will…" so that we can see an ideal interaction with the product.

You should provide contact information for two or three potential users of the product who can be reached by Development when clarification is required. A product manager is usually not a necessary interface between the developer and the customer; include phone numbers and email addresses so the developer will contact them directly.

- What is the persona doing to cause this problem?
- How does the persona do it now?
- How might the persona like to do it?

When questions arise about an implementation choice, the developer should find the answer in the use-case. Also, provide requirements and use-cases to QA for their test plans.

Sample Use Case: Michelle Makes a CD

Michelle owns 10 CDs that each contains only one or two "good" songs. She wants to copy the good ones onto a single CD to play in the car. With our product running, she inserts a CD. Since she's not already online with AOL, the program asks her if she wants to pull track names from CDDB. She does, so the program tells her to log on and then tell the program when she's ready. After she gets online, the track information is retrieved without further interaction with Michelle. Her display now shows track names and song lengths.

Michelle wants track 3 and track 6 on her mix CD. She drags each from the CD listing to a "mix CD" area. (She can ctrl-click or shift-click but probably won't.) The tracks names and times are now shown in the mix CD area. She continues this with additional CDs until the allocated time is exceeded.

Michelle plans to make a jewel case liner later so she hopes that the track info is stored somewhere with the disc info so she never has to type track names.

During the actual recording of the mix CD, she is relieved to see that she doesn't have to stay online nor does the program request the original CDs again (since she's forgotten which ones she selected and in what order).

Specification

Name. A unique identifier for a shorthand reference to the specification.

Description. In 50 words or less, describe how Development will respond to the requirement.

Difficulty. On an objective scale, how hard will it be to implement this specification?

Confidence. On an objective scale, what is the author's confidence that we can do this? For example, "we have done this before in another product" confidence = high. "We have to invent something that doesn't exist" should get a low confidence. Do not allow things with low confidence to be critical to the product.

Effort. (in man-weeks)[6]

Attachments. A prototype or sample screen shot is often attached.

Tracking code, author & document date (original and revised).

Information from the specification provides the basis for the costing estimates found in the business case.

Market Requirements in the Business Case

The collection of documents (Personas and Goals, Requirements and Use Cases) is collated into a Market Requirements Document, which is often appended to the business case.

A product manager owns the business case, which articulates the costs and revenue potential of a new product (or new release of an existing product). This also includes a summary table of requirements.

Executive Summary (one page)

- Vision/Theme
- Top 5 requirements
- Revenue potential
- Milestones
- Signatures of the key players

Market Opportunity

Market segment

Market sizing

Risks & Competitive issues

Market Requirements Table

Personas [User] (description)

Buyers [Technical and Economic] (description)

Requirements (table)

- Tracking code
- Name
- Type (functional, performance, constraint, interface)
- Category
- Source (Prospects, Users, R&D, Exec, Sales, Tech support, doc, mktg, Prod Mgmt)
- # of user inputs (prospect request/proposal, enhancement request, loss report)
- Phase/Release (based on persona goals, providing a roadmap of the key features for next few releases)

Additional information from the specifications:

- Difficulty
- Confidence
- Effort (in man-weeks)

Financial (multi-year)

Revenue potential

Costing/ROI

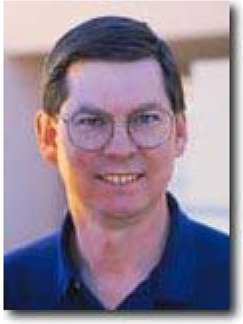
Marketing Plan (updated annually)

External to business case, includes channel programs, SWOT analysis, and costing

Document Control (revision history)

About the Author

Steve Johnson
sjohnson@pragmaticmarketing.com



Steve Johnson is an expert on product management and marketing in high-tech companies. He works for Pragmatic Marketing as an instructor for the top-rated courses "Practical Product Management" and "Requirements That Work" as well as onsite workshops.

Johnson has trained thousands of marketing professionals at hundreds of technology companies in the US and around the world. He is a top-rated speaker at many marketing and technology conferences.

♦♦♦

Learn more about Pragmatic Marketing by visiting http://www.pragmaticmarketing.com.

EndNotes

[1] "Product Management Roles and Salaries Survey" by Pragmatic Marketing. Download the complete survey from http://www.productmarketing.com/

[2] For more on buyer types, see *The New Strategic Selling* by Stephen Heiman. ISBN: 0446673463

[3] For more information on "personas" refer to *The Inmates are Running the Asylum* by Alan Cooper and Paul Saffo. ISBN: 0672316498

[4] Reference: IEEE Std 1220-1994

[5] "Characteristics of Good Requirements" by Pradip Kar and Michelle Bailey, given at the 6th INCOSE Symposium. Available at http://www.complianceautomation.com

[6] *Microsoft Secrets* by Richard W. Selby & Michael A. Cusumano explains how Microsoft estimates using weeks instead of months (too vague) or hours (so accurate that time is wasted). ISBN: 0684855313