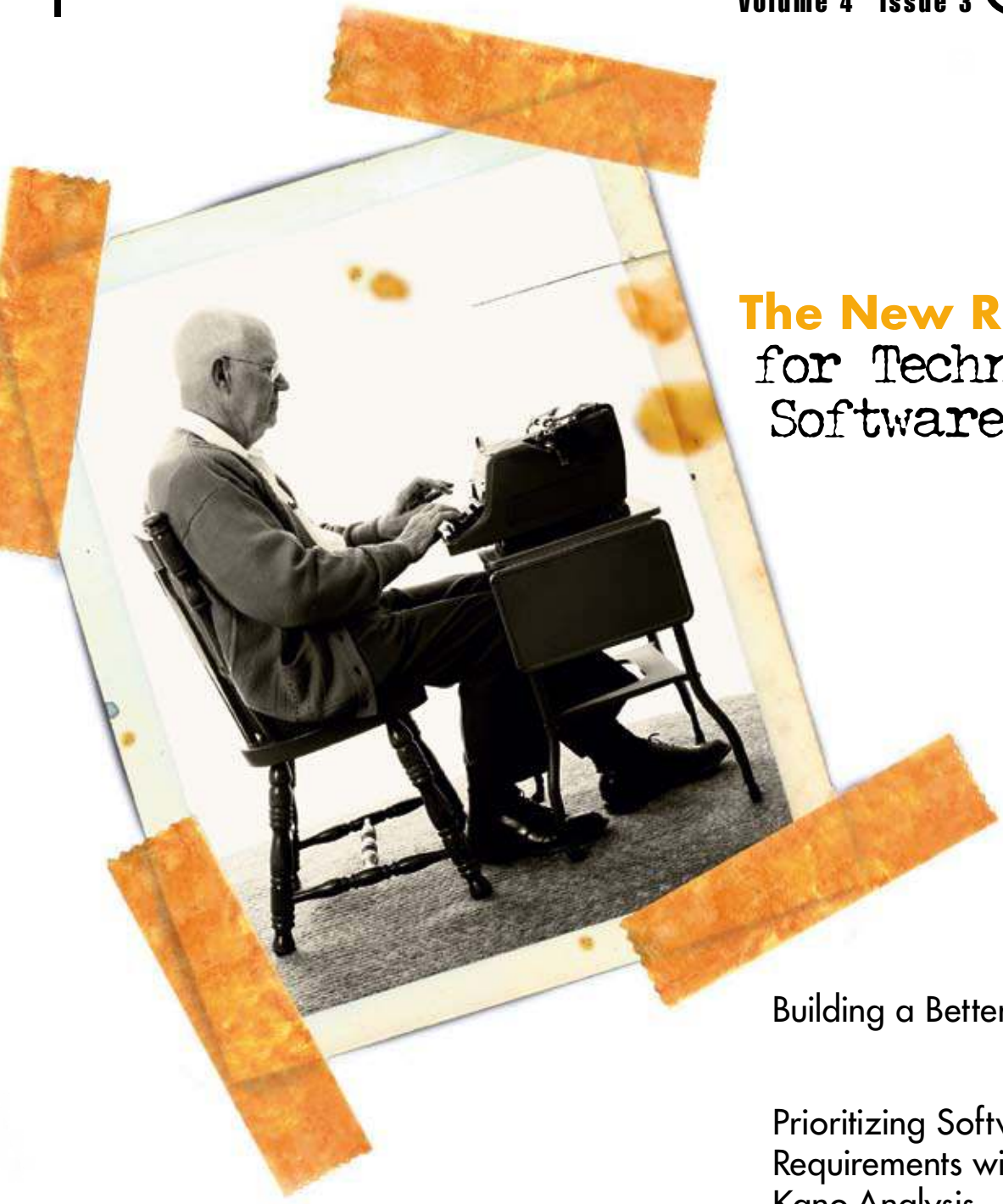
The Marketing Journal for Technology Product Managers • A Pragmatic Marketing® Publication

productmarketing.com™

Volume 4 Issue 3 2006



The New Rules of PR for Technology & Software Businesses

Building a Better Beta

Prioritizing Software Requirements with Kano Analysis

Thomson Medstat Transforms Product Management Through Intense Focus on Customer Needs

TURN THIS INTO THIS.

When it comes to product marketing frameworks, Pragmatic Marketing® is in a class of its own. Easy to use, straightforward, and highly effective, Pragmatic brings a best practices framework to the often-chaotic world of product management.

FeaturePlan™ takes the genius of Pragmatic and turns it into a powerful, automated product management tool for the software industry. It's Pragmatic for the desktop. And like Pragmatic, FeaturePlan is easy to use, straightforward and highly effective.

Produce product roadmaps. Ship software that's more aligned with market needs. See the power of Pragmatic at work—only with FeaturePlan.

If you've bought into Pragmatic Marketing, then you'll see why you've got to buy into FeaturePlan.

Automate.

Collaborate.

Organize.

Manage.

Communicate.

For more information about how FeaturePlan can help, contact us at **1-866-796-2832** or **sales@featureplan.com**

Visit us at www.featureplan.com
(and be sure to check out our 5-minute product demo)

Visit the Product Management Blog at
www.featureplan.com/community
Insights, Best Practices, webinars, and much more!

That's FeaturePlan.

FeaturePlan™

productmarketing.com™
16035 N. 80th Street, Suite F
Scottsdale, AZ 85260

President and CEO
Craig Stull
Pragmatic Marketing, Inc.

Managing Editor
Kristyn Benmoussa

Contributing Writers
Steve Johnson
Saeed Khan
David Meerman Scott
Scott Sehlhorst

Interested in contributing an article?
Email submissions to
kbenmoussa@pragmaticmarketing.com

No part of this publication may be reproduced, stored in any retrieval system, or transmitted, in any form or by any means, electronic, mechanical photocopying, recording or otherwise, without the prior written permission of the publisher.

productmarketing.com™ is available free of charge to qualified subscribers. For subscription or back issues call (480) 515-1411; or visit www.PragmaticMarketing.com/subscribe/

To be removed from the mail list, visit www.PragmaticMarketing.com/unsubscribe/ or send an email to info@PragmaticMarketing.com

For advertising rates, call (480) 515-1411.

Other product and/or company names mentioned in this journal may be trademarks or registered trademarks of their respective companies and are the sole property of their respective owners. *productmarketing.com*™, a Pragmatic Marketing, Inc. publication, shall not be liable regardless of the cause, for any errors, inaccuracies, omissions, or other defects in, or untimeliness or unauthenticity of, the information contained within this magazine. Pragmatic Marketing makes no representations or warranties as to the results obtained from the use of this information. Pragmatic Marketing does not warranty or guarantee the results obtained from the use of this information and shall not be liable for any third-party claims or losses of any kind, including lost profits, and punitive damages.

productmarketing.com is a trademark of Pragmatic Marketing, Inc.

Printed in the U.S.A.

All rights reserved.

About Pragmatic Marketing, Inc.

Pragmatic Marketing, Inc. was formed in 1993 to provide product management and marketing training to technology companies by focusing on strategic, market-driven techniques. Pragmatic's seminars emphasize a business-oriented definition of market problems, resulting in reduced risk and faster product delivery and adoption. Since its inception, Pragmatic Marketing has successfully trained over 30,000 product managers and marketing professionals. For more information, visit www.PragmaticMarketing.com or call (480) 515-1411.

Inside this issue:

Volume 4 Issue 3 • 2006

4 The New Rules of PR for Technology & Software Businesses

Because the rules for relating to the public have changed so slowly over the past ten years, practitioners who learned based on the old rules have been equally slow to change. It's time to step it up and consider the promise Web 2.0 public relations holds.

11 Ask the Expert

Does a formal requirements process stifle creativity and innovation?

12 Building a Better Beta

Delivering a successful beta program can be one of the hardest things to do as a product manager. Managing an effective beta program means doing what it takes to minimize the aspects that are out of the product manager's direct control and, like any other program that is executed, bring as much predictability to it as possible. Learn how planning and execution can be broken up into three phases, with each stage having several tasks and objectives.

24 Prioritizing Software Requirements with Kano Analysis

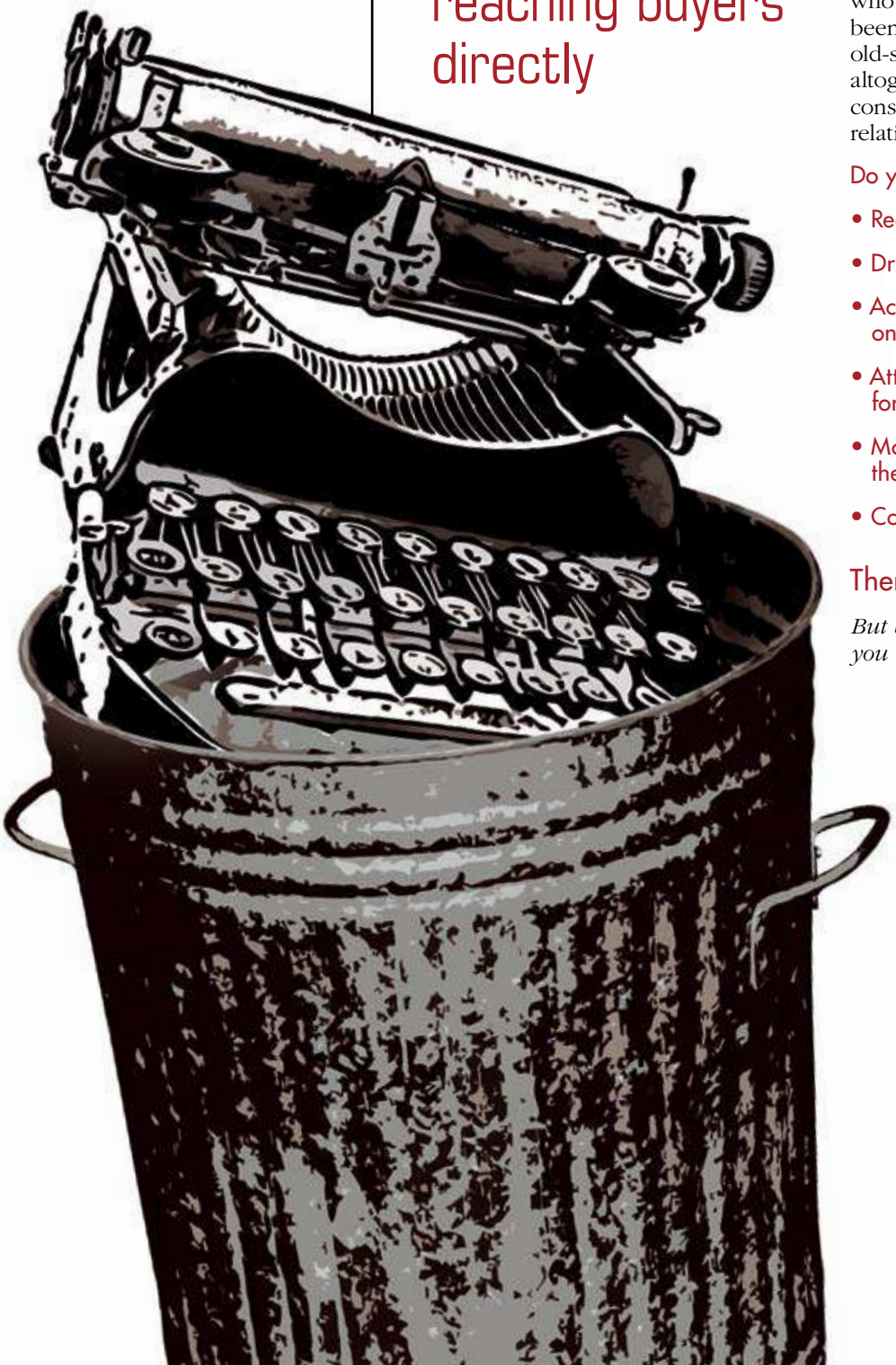
The Kano analysis model was developed to identify and contrast essential customer requirements from incremental requirements, and initiate critical thinking. Explore the four categories into which each feature or requirement can be classified.

29 Case Study

Thomson Medstat Transforms Product Management Through Intense Focus on Customer Needs

The New Rules of PR for

How to create a press release strategy for reaching buyers directly



The web has changed the rules for press releases. The thing is, most old-line PR professionals just don't know it yet.

Because the rules for relating to the public have changed so slowly over the past ten years (since the web has allowed people to read press releases directly), practitioners who learned based on the old rules have been equally slow to change. In fact, most old-school experts have refused to change altogether. It's time to step it up and consider the promise Web 2.0 public relations holds.

Do you want to:

- Reach your buyers directly?
- Drive traffic to your website?
- Achieve high rankings on search engines?
- Attract buyers who are looking for what you have to offer?
- Move people into and through the sales process?
- Compete more effectively?

Then read on.

But be prepared to alter the way you think about press releases.

Technology & Software Businesses

By David Meerman Scott

The old rules of press releases

In the old days, a press release was—shockingly—actually a release to the press.

Before the web, everybody knew that the only reason you issued a press release was to get the media to write about you.

Ye Olde Press Release Rules

- Nobody saw the actual press release except a handful of reporters and editors.
- You had to have significant news before you were allowed to write a press release.
- A release had to include quotes from third parties, such as customers, analysts and experts.
- The only way your buyers would learn about the press release's content was if the media wrote a story about it.
- The only way to measure the effectiveness of press releases was through "clip books," which collected every time the media deigned to pick up your release.

No More.

The web has transformed the rules and you must transform your releases to make the most of the web-enabled marketplace of ideas.

Why you need to learn the new rules

Today, savvy marketing professionals use press releases to reach buyers directly.

While many marketing and PR people understand that press releases sent over the wires appear in near real-time on services like Google™ News, very few understand the implications for how they must dramatically alter their press release strategy in order to maximize the effectiveness of the press release as a direct consumer-communication channel.

The media has been disintermediated. The web has changed the rules. Buyers read your press releases directly and you need to be talking their language.

This is not to suggest that media relations are no longer important; mainstream media and the trade press must be part of an overall communications strategy. In some businesses, mainstream media and the trade press remain critically important and, of course, the media still derives some of its content from press releases.

But your primary audience is no longer just a handful of journalists. Your audience is millions of people with Internet connections, and access to search engines and RSS readers.

The new rules of press releases

- Don't just send press releases when "big news" is happening; find good reasons to send them all the time.
- Instead of just targeting a handful of journalists, create press releases that appeal directly to your buyers.
- Write releases replete with keyword-rich copy.
- Create links in releases to deliver potential customers to landing pages on your website.
- Optimize press release delivery for searching and browsing.
- Drive people into the sales process with press releases.

OK, now that you know the new rules, you need to change the fundamental way that you use press releases. I'll share with you some specific strategies for leveraging the once lowly press release into one of the most important direct marketing tools at your disposal and, in the process, teach you how to drive buyers straight to your company's products and services.

Example of the new rules of press releases in action

Let me tell you a story.

I speak regularly at conferences and seminars. In the Fall of 2005, I was preparing a keynote speech called *"Shorten your sales cycle: Marketing programs that deliver more revenue faster."* To be honest, I was kind of procrastinating. Facing a blank PowerPoint® file, I decided to hit on Google in search of inspiration.

I entered the phrase "accelerate sales cycle" to see if there was anything interesting I could use in my presentation. The highest-ranked listings for this phrase on Google were from WebEx™, a company that provides online collaboration services. What was most interesting was that the links pointed to press releases on the WebEx site.

Then I went over to Google News and checked out the same phrase—"accelerate sales cycle"—and sure enough, WebEx also had the number one listing with a very recent press release dated September 28, 2005: "*Application Integration Industry Leader Optimizes Marketing and Sales Processes with WebEx Application Suite.*" The press release, about a WebEx customer, had been sent through PR Newswire and had a direct web link to the WebEx site to provide additional information.

WebEx also provided links in some press releases directly to free trial offers of their services. How cool is that?

WebEx provided me with a great example of a company that had optimized the content of press releases to include relevant terms such as the one I entered, "accelerate sales cycle." This is a terrific—and real—example of the new rules of press releases in action.

Reach your buyers directly

Under the old rules, the only way to get "published" was to have your press release "picked up" by the media.

We've come a long way. The web has turned all kinds of companies, non-profits, and even rock bands and political campaigners into just-in-time and just-right publishers. Organizations—the new publishers—create press releases that deliver useful information directly onto the screens of their buyers.

Until recently, nobody ever thought of companies as publishers; newspapers and magazines published the news. But that's all changing. Self-publishing web-style has moved into the mainstream and organizations large and small are doing the publishing… via press releases.

As you are making this fundamental shift, what should you write press releases about? Big news is great, but don't wait. Write about just about anything that your organization is doing:

CEO speaking at a conference?
Write a release.

Win an award?
Write a release.

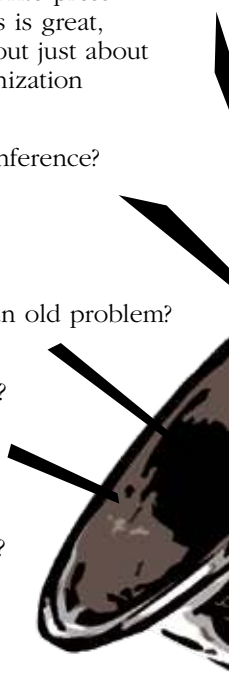
Have a new take on an old problem?
Write a release.

Add a product feature?
Write a release.

Win a new customer?
Write a release.

Publish a white paper?
Write a release.

Get out of bed this morning?
Okay, maybe not… but you are thinking the right way now!



Let the world know about your expertise

Every organization possesses particular expertise that has value in the new e-marketplace of ideas. The web has made it easy for organizations to publish that expertise in various forms, including press releases, which allows companies, institutions, and non-profits to function much like traditional publishers. Organizations gain credibility and loyalty with customers, employees, investors, and suppliers through content. Smart web marketers now think and act like publishers in order to create and deliver content targeted directly at their audience.

As organizations of all types begin to behave like publishers, many are adapting to the rigors of the publishing business and learning the editorial process. At the same time, new rules are emerging as online publishing continues to mature.

In an increasingly competitive marketplace, all organizations are searching for the elusive key to success. Well, look no further! Content, even in the form of a "mere" press release, will unlock success in almost any product category, even in highly competitive industries where smaller players are beset upon by larger, better-funded competitors.

Publish your press releases through a distribution service

Publishers also recognize that simply creating compelling content is not enough; it has to reach interested readers. The best way to publish press releases is to simultaneously post a release to your own website and send it to one of the press release wires.

There are a number of options for wire distribution of press releases. The benefit of using a press release distribution service is that your release will be sent to the online news services such as Yahoo!®, Google, Lycos®, and many others. Many press release distribution services reach trade and industry websites as well. In fact, you can reach hundreds of websites with a single press release. Take a look at the various services and compare them yourself.

A Selection of the Larger Press Release Distribution Services

Business Wire
www.businesswire.com

PRWeb®
www.prweb.com

PR Newswire
www.prnewswire.com

Market Wire™
www.marketwire.com

In order to get your press releases to appear on the online news services, you just have to get your release onto a basic press release "circuit" offered by a press release distribution service. The services also have many value-added options for you to consider. Compare options and in making your choice remember that your goal of sending press releases using the new rules is search engine marketing, so choose a distribution service that gets your releases to as many places where consumers see them as possible. Thus, the newsroom and geographical reach offered by a service is less important than ensuring that your releases are included on major online news sites.

RSS feeds from online news sites display your press release content

Many press release distribution services also offer RSS (Really Simple Syndication) feeds of their press releases, which they make available to other sites, blogs and individuals. What this means is that each time you publish a press release with them, the press release is seen by thousands of people who have subscribed to the RSS content feeds. And online news services such as Google News have RSS feed capability too, allowing people to receive feeds based on keywords and phrases. So each time your release includes a word or phrase of importance, people will receive your press releases directly and in near real-time.

Simultaneously publish your press releases to your website

Post your press releases to an appropriate and readily findable section of your website. Many organizations have a media room or press section of the website, which is ideal. The press release should be kept live for as long as the content is appropriate, perhaps for years. For many organizations, the press release section of the website is one of the most frequently visited parts of the site. Check out your own website statistics. You may be amazed at how many visitors are already reading your press releases.

When the press release is posted on your site, search engine crawlers will find the content, index it, and rank it based on words and phrases and other factors. To achieve high rankings, pay attention not only to the words and phrases, but also to other search engine optimization factors such as the URL used, the words in the headline and first paragraph of the release, the title tag, and metatags. Don't use image files for text, as the words cannot be read by search engine crawlers. You should also create your own RSS feeds of your news stories on your site. Use a standard off-the-shelf RSS feed generator so that interested visitors will be able to subscribe to your press release feed directly.

The importance of links in your press releases

Particularly because your releases may be delivered by feeds or on news services and various sites other than your own, creating links from your press releases to content on your website is very important. These links, which might point to a specific offer or to a landing page with more information, allow your buyers to link from the press release content into specific website content on your site that will then drive them into the sales process.

However, there is another enormous added benefit to including links in press releases. Each time your press release is posted on another site, such as an online news site, the inbound link from the online news site to your website helps increase the ranking of your site. This is because the search engines use inbound links as one of the important criteria for the page ranking algorithms. So when your press release has a link to your site, you actually increase the ranking of the pages that you link to—*yours*.

Focus on the keywords and phrases that your buyers use

First, determine who your target audience is and figure out how they should be sliced into distinct buying segments. Once this exercise is complete, identify the situations in which each target audience may find themselves. What are their problems? What keeps them awake at night? What do they want to know? What words and phrases do they use to describe these problems? The answers, by the way, may be very different from the way you would label them. Don't get trapped by your own jargon; think, speak, and write like your customers do. →

Drive people into the sales process with press releases

Savvy marketing professionals understand that Sales and Marketing must work together to move prospects into the sales pipeline. This is especially important in a complex B2B sale that has long decision-making cycles and multiple buyers that need to be influenced.

The good news is that press release content drives people into and shortens the sales cycle for any product or service, especially ones that have many steps and take months or years to complete. Here's how you can make this happen:

Understand your sales process in detail.

All sales processes are definable, repeatable, and understandable, and effective marketers use the web to move people into and through the process. You need to get together with salespeople, sales management and product managers to understand exactly what happens in the sales cycle. You should answer questions such as: How do people initially find your company or product? What words and phrases do they use? Understanding the process in detail allows you to understand how in the process press releases can be used.

Segment your prospects.

Consider press releases based on the buyer persona, perhaps by job title or by industry. Prospects are much more likely to enter the sales cycle by clicking a press release that talks about solving their problems.

Create thought leadership content to sit at the top of the sales funnel.

People in the early stages of the sales cycle need basic information on the product category, especially "thought leadership" pieces. Don't just write press releases about your company and your products; be a thought leader and write about the industry and higher-level strategic issues surrounding your product or service.

When doing initial research, people don't want to hear about you and your company; they want information about them and their problems.

Provide content that is compelling enough to get people to "raise their hand."

In your press releases, deliver something of value that you can trade for a registration form. A link from your press release to an informative white paper, online event (such as a Webinar), or online demo will help move your prospect further down the sales process—and, in exchange for the right content, the prospect will happily "raise their hand" to express interest by filling out a form. Remember, you're still not ready to sell a product or service (yet); you are still relationship-building. (By the way, these links will also help raise the search engine ranking of the pages you point to.)

Measure and improve.

Measure what content is being used and how. Understand through web metrics what's working and constantly tweak the content to make it better. Meet regularly with salespeople to gain insights into the sales cycle and how your press release content helps, and can better assist, the process. The web is iterative—you can and should make adjustments on the fly.

If the media does write about your organization, **GREAT!**

I have focused on how press releases are an effective way to communicate directly to your buyers. However, journalists still read and rely on press releases as they write stories. While you must work to craft releases targeted at your market target, your press releases should also make it into the hands of media professionals. The good news is that your efforts to communicate in a language potential buyers will find useful will not be lost on the media either; clearer, more effective press releases will make sense to them too.

And hey, if your releases generate a story about your organization, wonderful! Don't forget to link to it from your site and to purchase the online and print reprint rights for the article so you can promote it to your buyers, who may not have seen the story directly themselves.

Don't just wait for the media to talk. If you craft and deliver your press releases effectively, in today's web context, you will reach your constituents directly.

Prepare to be a maverick

If you've read this far, thank you.

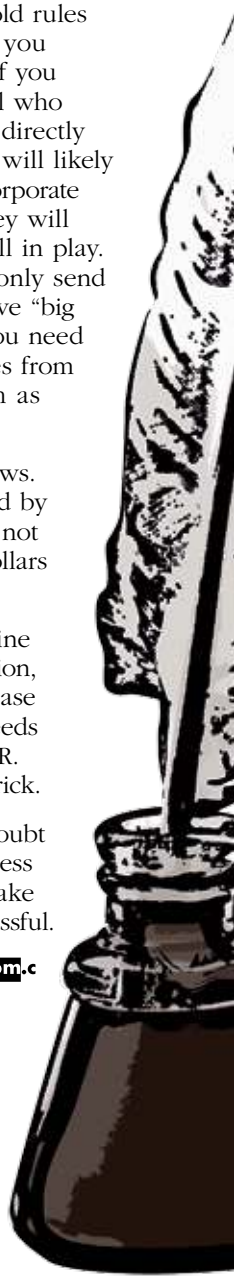
But now I must warn you: Many people who adhere to the old rules of press releases will fight you on executing this strategy. If you are a marketing professional who wants to reach your buyers directly through press releases, you will likely encounter resistance from corporate communications people. They will tell you the old rules are still in play. They will say that you can only send press releases when you have "big news." They will say that you need quotes in your press releases from respected third-parties such as analysts and customers.

But the old rules are old news. Press releases are being read by millions of people who are not journalists and billions of dollars of successful sales results.

If you face a stubborn old-line PR person in your organization, you may need to prepare a case for why your organization needs to adopt the new rules of PR. You will need to be a maverick.

Don't give up. There's no doubt that in the web economy, press releases drive action and make those who create them successful.

pm.c



So you want to go viral?

When I wanted to get the ideas on The New Rules of PR into the marketplace, I knew that the best way was to use thought leadership, a technique that is taught in Pragmatic Marketing's Effective Marketing Programs™ seminar. Many technology companies use thought leadership when they create white papers, Webinars, and podcasts or when executives speak at conferences. Rather than sell your company's products, thought leadership sells your organization as a smart player in the market that people will want to do business with.

Thought leadership is great, but one of the most difficult aspects is getting your white paper read or attracting people to your Webinar. I used viral marketing techniques to promote The New Rules of PR and smart technology marketers can learn from my experience.

In mid-January, I published the freshly formatted complimentary e-book *The new rules of PR: How to create a press release strategy for reaching buyers directly* on my site at www.davidmeermanscott.com/documents/New_Rules_of_PR.pdf and also on my blog at www.webinknow.com/2006/01/new_complimenta.html. I also sent a link to a few friends and colleagues. Since then, I have been happily watching the wave of interest from this little effort and I am amazed at the metrics. If the results that I've achieved are any indication, web content, particularly in the form of e-books are one of the best viral marketing initiatives that exist.

Within hours, hundreds of people had downloaded the e-book. And in the next several days, several alert bloggers picked up on the availability of the e-book and posted links. Over the next three days, the e-book generated just over 1,000 downloads per day. I was thrilled at the response. The power of a half-dozen bloggers telling the same story is tremendous. And how cool is it that on the web you can measure the results?

But that was just the beginning. Several days later, two-heavy hitters jumped in. Seth Godin wrote about the e-book in a post called "What would David Do?" Then Steve Rubell wrote on his blog, Micro Persuasion, a deliciously controversial post "Direct-to-Consumer Press Releases Suck." There's nothing like a negative viewpoint to drive viral marketing!

In just the next three days, 15,000 people downloaded the e-book. And with the wide reach of Godin and Rubell now adding to the momentum, several dozen other bloggers posted their thoughts on my e-book *The New Rules of PR* and since then more bloggers have picked up the thread.

The CEO of PRWeb (who I did not know) read the e-book and asked me to comment in a press release about trackbacks in press releases.

Here are some links and following trackbacks will lead to others:

http://sethgodin.typepad.com/seths_blog/2006/01/what_would_davi.html

http://www.micropersuasion.com/2006/01/directtoconsume.html

http://www.emediawire.com/releases/2006/2/emw349374.htm

As I write this, it is just a few months since the release of my e-book. When I first posted it, there was precisely one hit on Google for the phrase "new rules of pr." Two months of viral effects later there are well over 1,000 hits on Google all pointing to my e-book. Amazing. Over 60,000 people (and counting) have downloaded the e-book. Not bad for a little e-book that cost virtually no money to write, produce and promote.

When I felt that things were waning, I added to my own momentum by issuing a press release announcing that the e-book had reached the 50,000 milestone. www.prweb.com/releases/2006/2/prweb349854.htm. This effort yielded even more results. And to cap off this effort, MarketingSherpa named this campaign to the "Viral Marketing Hall of Fame 2006," one of 12 campaigns to be selected.

Think about how you can leverage some aspect of your company, market, or technology, then create a thought leadership piece (a Webinar, white paper, e-book, podcast, or something else) and then try to get the viral buzz going.

David Meerman Scott is an instructor for Pragmatic Marketing's Effective Marketing Programs seminar and is the author of Cashing in with Content: How innovative marketers use digital information to turn browsers into buyers. Check out his blog at www.WebInkNow.com

think with the grid.™

Pragmatic Marketing has always been focused on the unique challenges of marketing technology products and services. The framework we teach—refined and perfected over 20 years—highlights both "best practices" and practical ones to find and develop profitable opportunities, plan market-driven products, and create winning promotional and sales campaigns. Each seminar offers immediate actionable ideas, templates, and tools.

Visit www.PragmaticMarketing.com to learn more.

PRAGMATIC MARKETING®

The Industry Standard in Technology Product Management and Marketing Education.

Ask the Expert

"Doesn't a formal requirements process stifle creativity and innovation? Our company seems to spend more time writing requirements and specifications than we do actually writing code and developing products. Shouldn't we value the result more than the process?"

Who would start a development project without a clear idea of what the market wanted to buy? Oh wait! We do that all the time! But does the product become successful?

Requirements are merely a list of problems or features that are driven by the needs of the market. The negativity around requirements generally originates from two problems: 1) poor product management, and 2) a misguided view of innovation.

In the technology industry, innovation is the name assigned to "creativity" while requirements equates to "dull." Missing from both typical innovation and from formal requirements is context: What is the problem? Why does it occur? Who is having it? How do they deal with it now?

IDEO, a brilliant design firm, uses a method of innovation that begins with a clear statement of the problem and then field observation of it. Sounds like requirements to me.

Innovation is not about creating cool stuff in the isolation of a development lab. Innovation is not just adding more features. The real challenge of innovative projects is knowing what features to omit. The more we understand the persona and her use of the product, the more we will know what features are mandatory and which can be safely removed.

Railing against requirements is often another way of railing against poor product management. Too often, product managers are trying to document their opinions of a successful product. Students of Pragmatic Marketing® have learned that product managers are messengers for the market. They know that "your opinion, although interesting, is irrelevant." Product managers should bring market facts to the planning session instead of their opinions. And most of all, they should put those facts into context with problems, using scenarios and personas.

The relationship between product manager and development manager is like a partnership, a marriage. Each brings something to the relationship: The product manager brings market information; the development manager brings technical prowess. One handles the business decisions while the other handles the creative ones. And when conflict occurs—and it will—the answer can be found in market data rather than opinion.

One more point about market facts: We're not interested in what features customers say they want as much as an articulation of problems they have. Henry Ford reminds us that if he had asked the market what they wanted, they would have asked for a faster horse. Customers will ask for more of the same while continuing to struggle with problems. Product managers need to identify those problems, not ask for the customers' wish list of features.

Great product managers observe the problems and report them to Development—in the form of requirements.

Steve Johnson is an expert in technology product management. He works for Pragmatic Marketing as an instructor for the top-rated courses Practical Product Management® and Requirements That Work™ as well as onsite courses. Contact Steve at sjohnson@PragmaticMarketing.com

Building a better BETA

By Saeed Khan

Is it just me, or is delivering a successful beta program one of the hardest things to do as a product manager? The difficulty arises because so much is out of the product manager's direct control and in the hands of the beta testers themselves. And if you can't get motivated beta customers to provide you with detailed feedback, you can significantly diminish or delay the revenue potential for your product.

Managing an effective beta program means doing what it takes to minimize the aspects that are out of the product manager's direct control and, like any other program that is executed, bring as much predictability to it as possible.

I work in an enterprise software company, so our software is quite complex. It connects customers' corporate databases and information systems and is considered "mission critical" by a large majority of those customers. It is important to tell you this because as software can range from very simple to quite complex, so can beta programs. I have managed betas for smaller consumer applications as well as web applications, and while the scale and complexity of the beta programs has varied, they were always challenging to execute well.

Think about it. A beta program involves having people spend their valuable time using your unreleased, and in many cases, buggy software. And, on top of that, you usually have a fairly short period of time to gain valuable insights on what issues need to be addressed with the product before it is released.

What a beta is not

Before I get into the details of what a beta program is and how to optimally manage it, it's important to understand what a beta program is not. It's not:

- a substitute for good quality assurance
- a substitute for proper design and usability testing
- a tool for the sales team to help close deals in their funnel
- something that should be confined to the product management or product development groups in a company

First, a beta program is not a substitute for good quality assurance. Sending out an early release to beta testers and counting on them to find bugs that should be fixed is a very bad objective for a beta program. In fact, if your development or QA teams disagree with the previous sentence, then they should realize that what they are saying is that something which is fundamentally their responsibility should be outsourced to a group of external people. The next logical step would be to outsource the entire development or QA function altogether.

Related to the previous point, a beta program is not a substitute for proper design and usability testing. There are two major reasons for this. The first is simply that something that is an intrinsic part of the software development process cannot be left to beta customers to resolve. Efficient user interfaces need to be designed and implemented with an understanding of user needs. They must be based on easily understood cognitive models, support specific workflow patterns, minimize hidden or hard-to-access functionality, and in many cases, be adaptive or configurable to support a diverse pool of users. These things need thought, planning, and sufficient lead time to implement. And thus, if left until beta, almost guarantee that significant rewriting will be needed, possibly causing a delay in the product's release date.

Next, a beta program is not a tool for the sales team to help close deals in their funnel. I'm not saying that exposure to beta software can't help close a deal, but direct prospect participation in a beta program is a bad idea. Why not give a prospect early access to software to get them excited about what is coming down the pipeline? Well, first of all, you cannot sell beta software and there are easier ways to provide prospects information about what is coming in the next release. This can be achieved via on-site demonstrations, Webinars, or even white papers and presentations. Additionally, simple software products notwithstanding, the cost of supporting a beta site and the risk of exposing the prospect to potentially buggy functionality (it is not fully tested and debugged, after all) may outweigh any advantages or benefits in accelerating the software sale.

Finally, a beta program is not something that should be confined to the product management or product development groups. If a software company is working on a major product release, and particularly if that product generates a significant portion of the company's revenue (i.e. it is not a marginal product or a minor release), then people from across the entire company must work together to set objectives for the beta program and ensure they are met or exceeded.

After all, the beta program is almost always the first external exposure of the upcoming software product. And if this product is expected to drive future revenues for the company, then the value of, and investment in, the program must be looked at from an overall company perspective. Virtually every customer-facing group in the company should be involved in the beta program. Whether the goal is related to technical enablement for groups such as Support or Professional Services, identifying press and analyst references for Marketing, or accelerating time to revenue for Sales by developing early adopter sites; taking a broad, company-wide view of the beta will maximize the benefits of the beta program for the entire company.

Origins of beta programs

In regards to product development, the terms "alpha" and "beta" were coined by IBM in the 1960s to label product development checkpoints. Alpha meant a unit, module or component test phase and beta was the initial systems test. These terms came from even earlier A, B and C tests for hardware. The A-test was a feasibility test that was done before conducting any design or development. The B-test was a demonstration or prototype showing that the engineering model functioned according to specifications. And finally the C-test (which maps to the current usage of the beta test), was performed on early samples of the product being created.

Software built today is quite complex, and whether it is called beta testing, external validation, early release or something else, it is almost universally accepted that software development is an iterative process and that user feedback is critical to developing great software products. And to maximize the value of the acquired feedback, a structured process must be in place to manage the beta program from end to end. →

Beta program overview

Beta planning and execution can be broken up into three phases, with each stage having several tasks and objectives.

Phase I – Pre-Beta	Phase II – Beta	Phase III – Post-Beta
Defining beta objectives and the Beta Requirements Document	Launching beta	Final customer calls
Soliciting beta customers	Working with beta sites	Summarizing beta findings
Prioritizing beta customers	Recording and disseminating customer feedback	Addressing beta findings
Identifying internal resources and resource readiness	Providing technical support	Internal post-mortems
Educating internal teams and beta customers	Customer visits	Publish post-mortem results
Creating test case scenarios	Developing reference sites	
Beta software readiness	Extending beta	

Phase I – Pre-Beta

Defining beta objectives

As in most significant endeavors, planning and preparation are keys to success. With beta programs, success will depend heavily upon proper execution of activities before the beta testing actually begins. The first thing that must be done is to identify the goals of the beta. To do this, a number of questions must be thought through and answered.

What are the desired outcomes of beta? The main objective for beta is, of course, product feedback. But, what kind of feedback is needed? Are there specific functional areas of the product that require feedback? Is it focused on the user interface, performance, scalability, installation, upgrade, etc.? Are there certain configurations, platforms, use cases, or others scenarios that need to be tested? How many beta sites need to provide feedback for each of these configurations, platforms, use cases and scenarios?

For example, if the product is something like an application server, with new features related to scalability and transaction processing, as well as support for the latest versions of Linux®, you need to define how many beta participants you want testing these features, how many Linux beta testers you need, and also, as a control, how many beta sites you need testing on non-Linux platforms. This helps you identify whether there are any issues specifically related to the new platform. You should also see if existing customers can do some head-to-head performance and scalability tests comparing your old version with the new one.

Beyond product feedback, it is typical in enterprise software companies to solicit some beta participants as reference sites for press quotes and analyst briefings. How many of these reference sites are needed and what is the profile of those sites? Do they need to be direct customers or can they be partners? Do they need to be large, name-brand enterprises or can they be smaller, less well-known companies? Do they need to be in particular geographies around the world? If so, how many are needed in each geographic region?

Other questions to answer when planning for beta include defining the beta timeframe. When will it start, and when will it end? Assuming you have software that needs to be installed at customer sites, will customers install it or will you send someone there to install it for them? How will you distribute patches or interim builds to customers during the beta program? Do customers need training on the beta software? If yes, how will it be delivered? When will it be delivered? How will customers communicate their feedback to you during beta? How will they receive technical support? Will beta customers be allowed to communicate with each other? If yes, then how will you make that happen? These are key questions that should be discussed internally and agreed upon in the early planning stages of the beta program.

In a recent beta I managed for a major release of our company's flagship product, we discussed all of the issues above and decided the following:

- The beta would run for a period of three months, starting about seven months before the expected release date of the product.
- We would ship software to all customers on DVD media at the beginning of beta, but only provide patches on a one-off basis to those encountering serious problems while beta testing.
- For training, we would hold three 90-minute Webinars for the beta customers in the week after software was shipped. These Webinars would cover the installation process and all of the major functionality areas. We would also have live demos on these functional areas in the Webinars.
- We would set up weekly 30-minute calls with each beta site and assign a product manager and a program manager to each customer to manage communications and track their progress.
- Customers would use our standard customer support process to access technical assistance during beta and would use the weekly call to discuss higher-level issues such as specific product feedback.

Beta Requirements Document (BRD)

As you can see, there are many questions regarding the beta that need to be answered in the planning stage. All of this information should be collected and documented in what I call a Beta Requirements Document (BRD). The BRD is a single point of reference for all aspects of the beta. It contains information on the key internal stakeholders for the beta (i.e. from Product Management, Product Marketing, Customer Support, etc.), their responsibilities, key milestones, and process descriptions about the beta (i.e. how many customers are expected in the beta, their proposed profile, the beta registration and approval process, how software will be distributed, how support will be provided, how feedback will be collected and recorded, etc.). By creating and publishing the BRD, anyone within the company who needs an understanding of the beta program can read the relevant portions. It also becomes a reference point against which you can measure success after you have completed the beta.

Beta prospecting

The BRD should define the ideal profile of the participants you want to have in the beta. How many total beta participants are required? What mix of new and experienced users do you want? What mix of direct customers and partners are needed, etc.? But getting the right number and mix of participants to commit to beta is another

story. Remember, participating in a beta program means that people are going to spend their time on your unreleased software. Given peoples' busy work schedules, finding a significant number to actively participate during the timeframe of your beta is a challenge. Look at prospecting in the same way you look at sales: build a funnel of qualified prospects that is some multiple (e.g. two or three times) of the total number of beta participants you need, and then work to close as many of them as possible. And of course, to build this funnel, you will need to contact an even larger multiple of people.

As an example, in my recent beta program, we identified six major functional areas where we wanted beta feedback. We also decided that our goal was to have between three to five active beta participants per functional area. By "active" I mean people who are regularly using the beta software and providing meaningful and detailed functionality feedback. So, three to five active participants per area translated to 18 to 30 active beta participants overall. But ensuring that we met this number meant recruiting at least twice as many into the beta program.

There are always sites that initially commit to beta but then drop out or do not participate as actively as planned, so ensure that you have significantly more than the target number committed to beta. Assuming roughly a 50% drop off between those who commit to beta and those who actually participate, we decided we would need at least 40 to 50 fully qualified and approved beta participants at the beginning of the program.

We did a lot of prospecting to acquire these qualified sites. Over 500 customers and partners were contacted. We worked closely with our sales and marketing teams to identify prospects. We also canvassed customers we had visited in early requirements gathering trips for functionality in this release.

Part of our qualification process included understanding why the person or people were interested in participating in the beta. Was there a critical business problem that needed to be solved or did they just want to "kick the tires" of the new release? We also found partners, specifically VARs, SIs, partner ISVs, and even independent consultants to be particularly good beta candidates. Most had a clear business reason for wanting access to the beta and had staff that could be dedicated to the beta and work with us to provide very detailed product feedback.

With direct customers, the story was a little different. While we identified many well-intentioned direct customers, spending time on beta meant an additional task that had to be done on top of their full-time jobs. Thus, it was something that many of our beta contacts were not keen to tackle. In those cases, we worked with our account manager to escalate the benefits of the beta program higher up at the customer site and tried to get a management sponsor for participating in the beta. Overall, this was a very laborious process, and took about two months to complete.

At my company, Product Management recruits the beta sites, but product managers I have spoken with in other companies say that they have recruiting teams in their organizations who call and solicit beta customers, though in those cases, it is still Product Management's task to define the beta customer profiles and recruitment criteria. Keep in mind that recruiting is the most critical aspect of the beta program. Get it wrong—the wrong sites, the wrong mix of sites, or too few sites—and you are unlikely to meet any of the beta program's downstream goals. And while getting it right does not guarantee success, it does significantly increase the probability in reaching those goals.

Prioritizing beta customers

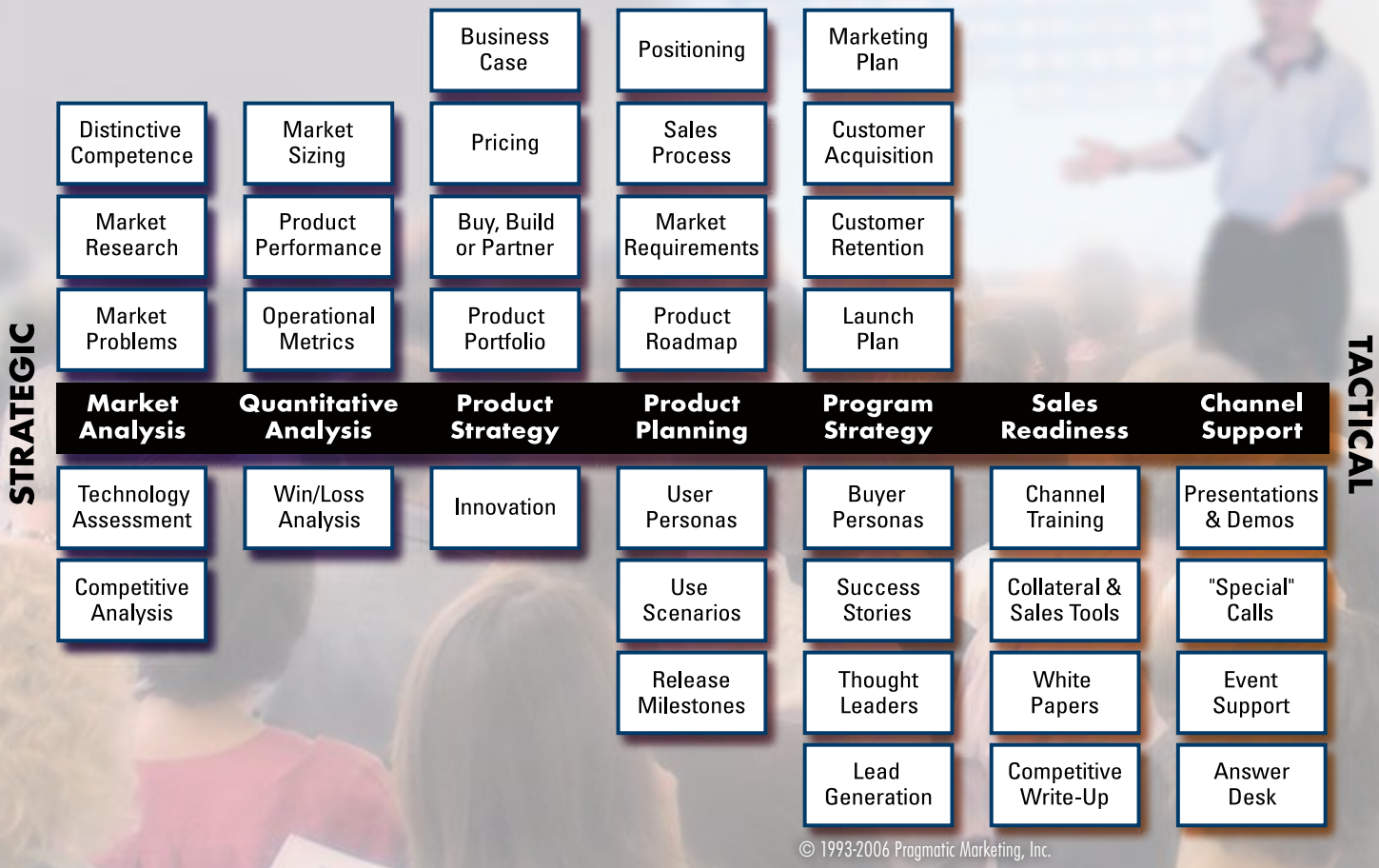
Not all beta customers are created equal. During the prospecting process, it will become evident that some customers are more likely to provide good feedback than others. As mentioned earlier, those with clear business needs addressed by the beta software, as well as partner companies, typically have the incentive to invest time and resources into your beta program. In addition to these companies, there may be some beta participants who are considered marquee customers and are potential beta reference sites. In short, a subset of the beta participants will warrant extra attention and support during the beta program as they are more likely than others to help you reach your targets.

Identify these premier companies early. Make all internal parties in your company who are involved in the beta program aware that these special firms will be monitored closely, and if needed, given additional help and support to ensure they are successful in the beta. Assign named technical resources to these sites to better ensure their beta program success. The product management team should work especially close with these customers, visiting them or engaging in additional contact to help resolve issues that arise.

You may not be able to accurately identify all premier sites until the beta is actually in progress. There are many things that can happen between the time a customer commits to participating in the beta and when the beta actually begins. In some cases, what looked like a promising site during registration turns out to be a dud once testing starts. And the opposite is also true. What sounds like a sleepy beta prospect turns into a very active and engaging beta site that far exceeds the original objectives. There is no harm in updating the premier list once beta starts, and in fact, it may become a necessity as the beta progresses.

(continued on page 18) →

Pragmatic Marketing Seminars



think with the grid.™

The recognized expert in technology product management and marketing, Pragmatic Marketing's common sense approach is built on years of experience and best practices. The Pragmatic Marketing seminars introduce a framework that gives technology marketers the tools necessary to deliver market-driven products that people want to buy. Our framework is the foundation for all our courses, from identifying markets and their problems to writing market requirements to creating go-to-market strategy that meets your corporate and revenue goals. Since 1993, over 30,000 attendees have been trained to "think with the grid."

PRAGMATIC MARKETING®

The Industry Standard in Technology Product Management and Marketing Education.

Curriculum for Technology Marketers

Practical Product Management®

Practical Product Management is for product managers and those who manage or contribute to aspects of product marketing and management. This two-day seminar fully explores the role of technical product management, providing tools and a framework to help get products to market more efficiently.

Complete seminar agendas are available at www.PragmaticMarketing.com

DAYS 1 – 2

I. Strategic Role of Product Management
- What is marketing?
- Definition of the role of product management
- Contrasting product management and product marketing
- Assigning ownership of responsibilities
- Identifying the "first steps" with gap analysis

II. Market Analysis
- Distinctive competence
- Market research
- Market problems
- Technology assessment
- Competitive analysis

III. Quantitative Analysis
- Market sizing
- Product performance
- Operational metrics
- Win/loss analysis

IV. Product Strategy
- Business case
- Pricing
- Buy, build, or partner?
- Thought leaders
- Innovation

V. Product Planning
- Positioning
- Sales process

VI. Case Study

VII. Delineating Responsibilities
- Communicating market facts to Development, Marcom, and Sales
- Drawing the line between Product Management and the other departments

Still not sure Practical Product Management is for you?

Attend a **FREE** 1/2-day session. See back cover for details.

Requirements That Work™

Requirements That Work is an intensive one-day course that introduces a straightforward method for creating product plans that product managers can write, developers readily embrace, and that produce solutions the market wants to buy.

Complete seminar agendas are available at www.PragmaticMarketing.com

DAY 3 – Requirements That Work *(For those who write requirements)*

VIII. Building the Market Requirements Document (MRD)
- Writing requirements
- Implementing use-case scenarios
- Programming for the "persona"
- Determining product feature sets
- Creating the MRD

IX. Analyzing Business and Technology Drivers
- Reviewing specifications
- Prioritizing the product feature set

X. Getting (and Keeping) Commitments
- Product contract
- Getting the product team in sync
- Getting executive support
- Communicating the plan in the company and in the market

Visit www.PragmaticMarketing.com or call (800) 816-7861 to register

Recruiting internal resources

As beta recruiting is taking place, you need to ensure internal resources are identified and will be prepared for beta. Who will provide technical support to the beta sites? Is it individuals from the technical support organization, from professional services, sales consulting, or could it even be from your development and quality assurance teams? It varies across companies, typically depending on the nature of the software and the size of the company. For smaller companies, beta support may be delivered by Engineering, whereas in larger companies, the customer support team may handle it.

Internal education

Once you have defined the internal team members, ensure appropriate individuals—particularly those who will provide the technical support for the beta—are properly educated on the beta software. Depending on the complexity of the product, this could be as simple as giving the internal groups access to the software and having them educate themselves via the documentation, or it could range all the way up to delivering formal training classes ahead of the release of beta software to customers. The objective ensures that your internal teams are at least one step ahead of customers, so that as customers start installing and using the software, the internal teams are ready to respond to any problems.

Educating beta sites

Beta customers typically need some form of education or training before using the beta software. If the software is a consumer-oriented product, then ideally, little or no training should be required. Worst case, the customer should be able to read or view a short online tutorial to get them going. Conversely, complex business or enterprise software requires extensive training. As mentioned earlier, we held three 90-minute Webinars for our beta customers. Virtually all beta customers attended all three Webinars and we received positive feedback from the attendees. While that made us feel good, we also knew that Webinars would only partially prepare customers to use the product and we would have to help them by explaining functionality during the beta itself. We also provided documentation with the beta, but that too was in a "beta" state, and while most sections were reasonably complete, more detail could and would be added later in the development cycle.

Providing test case scenarios

Without some guidance, a lot of beta feedback can be vague and generic. Statements like, "I tried this, this and that, and they all seemed to work OK," or "It seems to work as I would expect, but I noticed the buttons in the dialog box are not properly aligned.", are not uncommon when testers are left to their own devices. To help guide the testing done at each site, provide the beta customer with some structure. For example, define beta test scenarios for each area of functionality that needs to be tested. The purpose of these scenarios is to provide a clear framework to the customer in which to test the software. It is not to provide a step-by-step set of actions that they should perform.

The scenarios should focus on typical use cases you expect customers to perform once the software is released. The scenarios can also focus on specific aspects of functionality that need testing in customer environments. Do not expect beta customers to be able to perform in-depth stress testing of your software. This can be rather time consuming to prepare for and execute, and most beta customers will not have the time and resources to do this level of testing. The scenarios should be made available at the beginning of Phase II and can be introduced during the beta Webinars or in the first customer calls once beta has begun.

Beta software readiness

Another important task in preparing for beta is to track the readiness of the beta software. There are two major questions to answer. How can you measure if the software is ready to ship when the beta begins, and what do you do if you determine that it is not ready? It comes down to whether you want to be date-driven, quality-driven, or in most cases, an acceptable mix of the two.

Determining shipping readiness means measuring the quality of the software and deciding if it meets preset, agreed-upon criteria. If it meets the criteria, it is ready to ship. If not, more work needs to be done on it. Software quality can be an elusive concept, so how does one measure it?

The first thing that comes to mind is the number of open bugs and their severity. A lot of discussion will take place in Development or QA meetings about the number of critical or showstopper bugs that are in the current release. Many times statements are made that sound something like, "We cannot release the beta until all critical bugs are fixed and there are less than X number of severe bugs.", where X is some arbitrary, but reasonable sounding number. The problem with a statement like this is that, in reality, it is very difficult to enforce.

First, as the target beta date approaches, and if the number of critical and severe bugs remains high, the definitions of both critical and severe will magically loosen. What was once a critical bug two months prior to beta becomes a severe bug two weeks before beta. And what was once a severe bug, may become a moderate bug.

Second, what if all but "a few" critical bugs can be fixed by the beta date, will you still ship the beta? What if, instead of X severe bugs, there are 5% more than X severe bugs, will you ship it? You can always issue a patch or update of the beta software later, can't you?

Third, the statements about number of bugs are one-dimensional. The statements treat all critical bugs the same and all severe bugs the same. In reality, different functional areas need to be tracked separately with metrics for each area. In some cases, there may be lower tolerance levels acceptable for bugs, in other cases, it may be higher. Overall, things like usability, performance, installation, and upgrade path need to be measured and tracked separately with their own criteria and requirements.

For example, if your software has an installer, the installer *must* be bug free. If the installer is buggy, not only will the initial user experience be negative, but in cases where the user encounters a bug, the user will either have to call Support for help, or perhaps worse, abandon the install altogether. If you don't track this as a separate item and only look at aggregate bug numbers of different severities, then installation bugs will come back to haunt you.

Usability issues are another facet of applications that must be tracked separately. Many software Graphical User Interfaces (GUI) do not pass through rigorous design, review and implementation processes. They may be implemented by engineers, who may not be GUI experts, and then modified later as issues are raised. The GUI is the face of the product, and the best server or back-end software can only be hampered by a difficult GUI.

With respect to GUIs and beta programs, remember that:

- The beta is usually the first exposure the public has to the software and they will form opinions of the software based on the beta. A significant part of that opinion will be based on their emotional experience with the GUI.
- Many of these same people will be candidates for marketing reference sites. If their emotional experience is negative, they will likely not want to be associated with the product, even if you tell them that everything will be "fixed" when the software is released.
- Usability reviews should be performed on the pre-beta software and a commitment made to address the most important issues before the beta. The completion of those issues becomes one metric used to measure release readiness.

All of the readiness criteria should be converted to a number of metrics that are tracked and reviewed via a management dashboard. This dashboard shows high-level status for each category (e.g. via red, yellow, and green indicators) with the ability to drill down into details where needed. If available, you can use a dedicated reporting or dashboarding tool. But if not, then virtually any application, such as a spreadsheet, presentation tool or word processor will do. Regardless of how it is implemented, there are a number of benefits to this approach.

A management dashboard provides a top-down view of the state of product quality. A lot of red on the dashboard a few weeks before beta is set to launch may give you early warning that you will need to slip the release date or that additional focus or resources are needed to meet your existing dates. If most areas show green, but a few are persistently red or yellow, it may indicate issues with the teams working on those areas of functionality, and specific action can be taken to help those team members. Another obvious benefit of such a dashboard is that it gives upper management clear visibility into the status of the software and the ability to make better and quicker decisions to reallocate resources to where they may be needed.

Phase II – Beta

Launching beta

After all of this preparation, you are now ready to deliver the software to customers and begin the actual beta testing. In most cases, customers should be able to install the software on their own. If the installation is complex, you can either send someone onsite to help install the software, or provide preinstalled software on a server or desktop machine. Keep in mind that while it may be possible to provide extra installation assistance during beta, this is most likely not going to be the case when the software is generally available. So, if the beta requires extra effort to install and configure, make sure you know why that is the case, and have a very clear and firm plan to address it before final software is released.

Working with beta sites

The major aim of the beta is to collect feedback from the beta sites. Despite any assurances to the contrary from beta sites, most sites will need a lot of prodding and assistance during the beta program if you want their substantive feedback. Based on my own beta experiences, combined with some anecdotal data from other enterprise software vendors, roughly half of all beta sites (50%) will end up installing and using the software in some way, and roughly half of those (25%) will provide meaningful feedback on at least one area of functionality. Thus, if you start with 50 registered beta sites, approximately 20 to 30 will actually install the software and provide some level of useful feedback, with about 10 to 15 giving detailed and very meaningful feedback. Of course every situation is different, but these ratios have held true for me across several beta programs. →

Regardless of the number of active beta sites, there are many ways to collect feedback. Depending on your goals, you can use the following techniques:

- Weekly calls with individual participants
- Regular conference calls with multiple participants
- Beta participant blog and/or discussion forums
- As needed email and phone conversations

Weekly calls with individual participants work very well. Although it can be time consuming to hold weekly meetings with each beta site, customers will commit to a weekly 30-minute call, and in some cases a 60-minute call, to discuss their issues and feedback. The benefit of the weekly call is that the beta customer can block off a fixed amount of time each week and knows they will get your undivided attention for that period of time. If they have problems with the beta software, the weekly call creates an opportunity to discuss the issues and get the customer back on track. If they are making progress with the beta, this weekly call gives you the opportunity to discuss their feedback in great detail and guide them further along in the directions you want them to go.

In addition to weekly calls with individual beta customers, you can set up regular, perhaps biweekly, conference calls with multiple customers. There is a risk of these calls devolving into a gripe session—it only takes one unhappy beta customer to get a group started—but if managed well, listening to customers discuss their issues with one another can lead to some valuable insights.

Another mechanism for communicating with beta customers is to provide a blog or web-based discussion forum to enable them to ask and answer questions, post findings or simply discuss relevant issues with one another. The benefit of providing an electronic forum is that the information is automatically persisted for you, and if supported by the software, customers can upload files, images, reports or any other documents that are related to the beta and their findings. These web-based forums can also be used as customer self-help tools, enabling beta customers to ask and answer questions amongst themselves.

Last and—in my opinion—least, let the customer contact you via email or phone on their own schedule and not set up formal weekly or biweekly calls. While a dedicated beta customer may be quite proactive and eloquent if left on their own, in my experience, leaving it up to the beta customer to contact you when they can will lead to little, if any, meaningful feedback.

Recording and disseminating customer feedback

Collecting customer feedback is one thing, but recording and disseminating it within your organization is something completely different. Remember that different internal groups require various levels of detail of the beta feedback. Obviously the most detailed information is in the form of the raw notes and findings taken from weekly customer calls. This raw feedback should be posted in an internal forum, such as on a wiki or portal, so that all parties can access it easily. The main consumers of this level of information will most likely be engineers and product managers. Raw feedback contains various types of information about customer issues, positive and negative comments on various parts of the software, access to technical support, plans, expectations, etc.

The information collected from these calls should be sliced and diced, and presented in multiple ways so that it can be efficiently utilized. Information should be available in the following ways:

1. Weekly call notes by customer (i.e. all calls for individual customers)
2. Customer call notes by week (i.e. all calls for a given week)
3. Detailed feedback by area of functionality
4. High-level summary of beta findings
5. Overall beta customer status by week

There may be other ways to present the data, but those listed above will give most interested parties what they need. Account managers will be very interested in the weekly progress of their accounts. Product managers and development managers will focus on the detailed feedback by area of functionality. Overall beta customer status by week is interesting to track. As customers receive software or begin their beta participation, it is important to track a high-level status for them each week. This status should be based on clear indicators of their progress during the course of beta. A set of statuses, including registered, received software, installed software, testing in progress, blocked, and completed beta, applied to each beta customer each week can be a valuable tool in tracking issues and progress during the beta. These types of simple metrics can be used to create dashboards or indicators so senior management can get a clear picture over time of the progress and status of the beta program.

For example, if early in the beta program a significant number of customers are blocked by installation or configuration issues, this will be clearly visible in the dashboard. Corrective action can be taken, but if the issue persists, decisions can be made to either resend a newer version that installs correctly, provide additional on-site help if warranted, or even consider extending the beta to make up for lost time.

Providing technical support

Most software companies have well-defined policies and procedures for delivering customer support. Typically each customer has some form of maintenance agreement with the vendor. But during beta programs, where even small issues can block customer progress, there may need to be a simplified and expedited mechanism for customers to log issues that need to be resolved. I have heard many

customers request this during beta programs. This could mean that the normal process to log and troubleshoot cases is too cumbersome, and while that may be true, it can also indicate that customers are not willing to invest as much time to resolve beta issues as they are regular software issues.

An expedited process should be set up for beta. It can be as simple as a dedicated email address for customers to use to indicate a problem. If one needs a little more sophistication, it is very simple to set up a structured web form that customers can use to submit issues. Another idea is to set up a shared discussion board for beta customers so they can log issues. Resolutions to common problems can also be posted on the discussion board. Sometimes, a beta customer who logs one problem may be able to offer a solution to another. This way, not only can some of the load be taken off of the customer support team, but beta customers can achieve a level of self-service in resolving problems.

Customer visits

As every product manager knows, there is nothing like a visit to a customer site to get a deeper understanding of customer issues. Additionally, customers often look forward to visits from product managers and will put aside several hours of their time, if needed, to discuss their concerns. So what better time to visit customers and discuss issues and concerns than during a beta program? Part of the beta planning process should include budget and goals for customer visits. The primary goal of the visits is to get more detailed information about the beta experience of the site, but do not limit it to that. The visit is a great opportunity to discuss future plans for your product in their organization. You can also uncover new projects or initiatives where your product can be used and even meet other people in the organization who can influence the use of your product in the company. Be very careful to keep the tone of the visit as a research visit and not turn it into a sales or prospecting call. As a product manager, you have a lot of credibility and influence with the customer, so don't jeopardize that by focusing on sales objectives instead of the customer needs.

Developing reference sites

In my experience, there is no direct relationship between someone having a positive beta experience and their ability to become a reference site. Certainly a positive experience improves the chances of someone becoming a reference, but there are numerous factors that can prevent that. Many companies have strict policies about their employees speaking publicly on behalf of a vendor. For those companies that allow their employees to speak publicly, there are usually strict guidelines as to what they can say and to whom they can say it. Those who do speak usually do so because they have a very good personal relationship with individuals at your company, and do not necessarily need to be actively involved in a full-blown beta program to assess the product and determine their opinion. Perhaps a live demo or a customer day at their site can be sufficient to get them acquainted and comfortable enough with your product to speak about it.

The point is that you should not depend *solely* on the beta program to develop press and analyst references for your product launch. Certainly, some beta sites can turn into references if your beta program progresses well, and those who have good beta experiences also have the right profile, authority and permission to speak on behalf of their company about your product. But, that leaves far too much to chance. What if your beta program is not successful, or what if very few beta sites have a good experience? What if those sites that do have a good experience with the beta do not fit the profile of type of reference site you want? What if those who do fit the profile and have a good experience do not have the permission to speak publicly?

To mitigate these risks, you need to have a separate but parallel effort to identify potential reference sites and work with them to get them to a point that they will become references. This effort is typically performed by Marketing or Product Marketing and not by Product Management. As mentioned earlier, the reference prospects may already have a personal relationship with individuals in your company and may not need full beta exposure to gain the product knowledge they need in order to speak about it. If some of these potential reference sites do require beta participation before becoming references, then let them participate. But once they enter the beta, all interaction with them should be handled with the full knowledge that they are a potential reference site, and not a site that will necessarily provide detailed feedback on features and functionality.

Extending beta

Depending on how your beta program progresses, you may want or need to extend the time period over which you hold the beta. Sometimes customers start late in the beta cycle and want additional time to provide feedback or the beta program identifies a number of issues in the product, and you realize that those issues are important enough that more product feedback is needed on the product. Regardless of the reason, it is important to know that this is a possibility and to prepare for it in advance. If you embed a time-bound license with your beta software, make sure that you can update that license if you extend the beta. If you cannot extend it after the fact, build additional time into the original license so that if you extend the beta, those customers who do continue to use the software for the extended period can do so without interruption. Also, make sure that the internal resources such as Technical Support and Professional Services who will be assisting customers in the beta are available beyond the initial beta period, if needed. Their managers may not be aware of the potential extended timeframe and may have other plans for them once the regular beta period ends.

Phase III – Post-Beta

Final customer calls

During Phase II of the beta, there is regular customer contact focusing on progress with the beta, with the supplied use cases and any issues that the customer faces with the product. This is typically tactical information focused on the current week or stage of beta customer participation. To close out the beta program with customers, you should hold a final wrap up call to discuss higher-level issues and obtain feedback on the overall beta program. This information is extremely valuable in assessing overall beta success and to identify the do's and don'ts for future beta programs. Below is a short list of questions I asked customers at the close of a recent beta program. Note that these are discussion points, and it is up to you to drill down with the beta customer to draw out additional details, where needed.

- Did the Webinars provide sufficient information to help you understand the capabilities of the beta software?
- Were the weekly conference calls helpful to you?
  - –What was helpful about them?
  - –What was not helpful, and what could we improve?
- What could be done to make the time on the calls more productive?
- Did you find the process of requesting support intuitive and straightforward?
- Did the technical support process meet your needs?
- How could support be improved for future beta programs?
- Did you have adequate hardware to test out the beta?
- Were you able to easily get up and running with the beta software?
  - –What could be done to make the process easier?
- For each area of functionality tested, what were the standout positives and negatives?
  - –What changes/enhancements are critical to the initial release of the product?
- Do you plan to upgrade to the new release? If so, when? If not, why not?

Given the detail of some of these questions, you should schedule at least one hour for these calls and ideally send the questions in advance to the beta customers. Getting the questions in advance will let them collect detailed and accurate answers to these questions and, if needed, poll their coworkers who also participated in the beta program.

Summarizing and addressing beta findings

Once the beta is over and the final customer calls are wrapped up, all the findings need to be consolidated and summarized into product and process categories. The product feedback from the beta must be converted into specific requirements that need to be addressed. The critical requirements can be added to your overall release criteria for the product. That is, the software cannot be released to market without addressing these critical needs. The remaining requirements can be scheduled for future feature or maintenance releases, and their priority can be assessed in the context of other requirements for those releases.

Internal post-mortem sessions and publishing results

The process objectives also need to be understood and implemented. But unlike the beta feedback that will come almost exclusively from beta customers, the process objectives have both an external as well as internal component. To obtain the internal beta process feedback, beta post-mortem or review sessions need to be held. These sessions should be led by Product Management and be held with individual teams such as Technical Support, Marketing, Quality Assurance, and Development. The reason to hold them with separate teams and not in one large session is so the discussion can remain focused and can drill into detail on the issues for each group. Once the individual sessions are held, the findings from all the groups can be combined and presented back in a joint meeting to the internal teams. This final meeting helps bring all groups together and everyone can collectively discuss what should and should not be done in future beta programs.

Conclusion

Executing a successful beta program requires approaching it in a very analytic manner. You must focus on minimizing the inherent uncertainty in collecting feedback on a pre-release product from very busy customers and partners. No one will argue that beta programs are a key element of the software development process, but planning for and executing a successful beta program is a difficult task. By taking a rigorous and methodical approach, and working to make each individual beta customer successful, you increase the likelihood of meeting your goals, delivering necessary market feedback to your organization, and significantly enhancing the chances of success for your product.

Saeed Khan is a 20-year veteran of the software industry with deep experience in product management, product marketing, and education. Saeed has managed market-leading products in a number of segments including data integration, data visualization and enterprise application development. Saeed is based in Toronto Canada, and can be reached at saeed_w_khan@yahoo.com

Problem.

Sales support and tactical marketing are consuming too much of your time.

Solution.

Put some strategy and process into your outbound activities.

Several thousand technology company marketers* have attended Effective Marketing Programs to learn how inbound and outbound marketing can work together to create a competitive market advantage for sales. They're telling us that we've got the right answers.

We won't promise to eliminate all of your problems overnight (we are pragmatic people, after all). But if you can break away for our two-day seminar, we'll show you how to convince both your internal and external audiences that your solutions deserve their attention. Just think of all the time you'll have for your real job!

Visit **www.PragmaticMarketing.com/emp** to learn more about the problems we solve at the Effective Marketing Programs seminar and download *"Strategic Marketing as the Source of Competitive Advantage."*

*Effective Marketing Programs is recommended for product management, product marketing, industry marketing, field marketing, sales operations, and marketing communications professionals.

"Effective Marketing Programs generated immediate benefits for my organization."

Robert Thomas
Division Vice President Marketing
ADP – Time and Labor Management

Effective Marketing Programs™

Pragmatic go-to-market strategies.

Visit www.PragmaticMarketing.com/emp
or call (800) 816-7861 to register

Pragmatic Marketing®

The Industry Standard in Technology Product Management and Marketing Education.

Prioritizing Software Requirements with

Noriaki Kano developed the Kano analysis model in the late 1980s to identify and contrast essential customer requirements from incremental requirements. One of his goals was to initiate critical thinking about the nature of requirements. His characterization approach can be used to drive prioritization of software requirements.

Kano analysis allows us to prioritize requirements as a function of customer satisfaction.

Kano defined four categories into which each feature or requirement can be classified (an Apple® iPod® is used for examples in each of the following four requirement categories):

1. ***Surprise and delight.***
   Capabilities that differentiate a product from its competition (e.g. the iPod nav-wheel).
2. ***More is better.***
   Dimensions along a continuum with a clear direction of increasing utility (e.g. battery life or song capacity).
3. ***Must be.***
   Functional barriers to entry—without these capabilities, customers will not use the product (e.g. UL approval).
4. ***Better not be.***
   Represents things that dissatisfy customers (e.g. inability to increase song capacity via upgrades).

Surprise and delight requirements

For just a moment, think about software as a user, not an accountant. We want software that is interesting and fun to use. Affordances in the user interface that allow us to just "do what comes naturally" and have the software do exactly what we want. New ideas that make software better. We're not talking about a button that pops up dancing squirrels when clicked, rather valuable features that make software great.

Great examples of valuable features include:

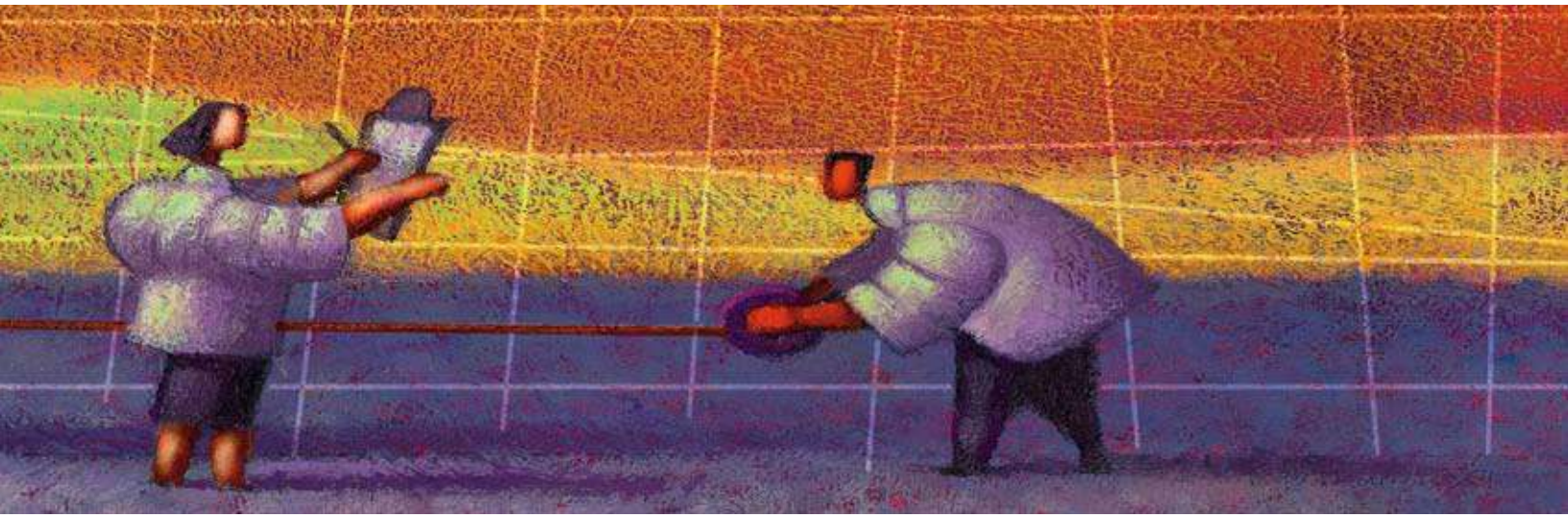
- The nav-wheel on the iPod, as a good hardware example.
- Google's Gmail™ use of labels instead of folders for organizing email, as a good software example.
- Contextual help buttons that open to exactly the right page in a Help file.

All of the examples above are *implementation details* or the results of design decisions—not part of specifying requirements. However, when converting from market requirements to product requirements, we can point our development teams in the right direction and help them focus on innovative solutions to the right problems. These might be the requirements behind the delightful features listed above.

- Users must be able to select songs while holding the iPod in one hand.
- The system must provide an efficient way to organize email, with the assumption that users will never delete email.
- The system shall provide relevant help information for the context in which the user requests help.

Kano Analysis

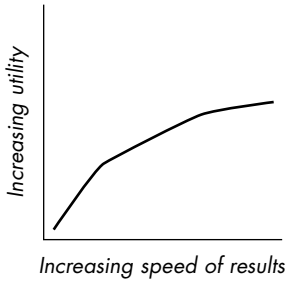
By Scott Sehlhorst



More is better requirements

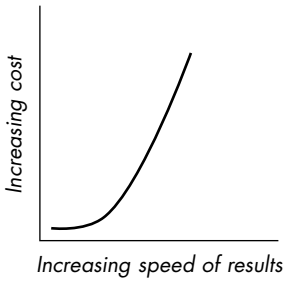
These are the most easily grasped concepts—bigger, faster, better, stronger. The challenge in writing a *more is better* requirement is in knowing when enough is enough. Requirements such as "minimize" or "maximize" are ambiguous. What is the theoretical minimum response time for a search engine? Does it take a few hundred micro-seconds for the bits to travel from the server to the user, plus a few micro-seconds for switch latency, plus a few nano-seconds for a CPU to find the answer? It would be completely impractical to unambiguously request that our developers minimize search time.

Specifying precise objectives can be very difficult as well. The law of diminishing returns comes into play. There is a concept in economics called *utility* which represents the tangible and intangible benefits of something. We can consider the utility of a feature with respect to the target users. A graph of the utility for *speed of search-result generation* would look like this:

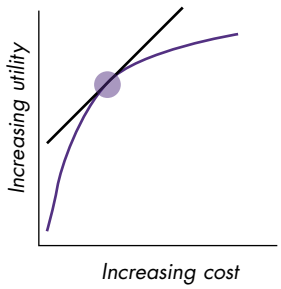


We can see that as the speed of results increases, the incremental benefit to the user decreases. While utility is strictly increasing, it is increasing by less and less. When writing a requirement, how do we determine the speed that is truly required? It would be ambiguous to say "as fast as possible" or "as fast as is reasonable." And it would be naive to think that we did not need to understand something about the implementation before specifying an unambiguous requirement like "search must complete in 2 seconds."

Thus far, we have only described the *benefit* side of the *cost-benefit analysis* needed to specify the requirement. We have to *iterate and interact* with our development team to determine the impact of a speed specification on costs. After getting feedback from our implementation team, we now have an understanding of the cost of implementing "search," as shown below:



We can see how it gets progressively more expensive to make progressively smaller increases in speed. This is our "development reality" and we can't ignore it when specifying how fast "search" needs to be. To determine the optimal specification, we have to find the point in the curves where the incremental benefit of searching faster is equal to the incremental cost of searching faster. We can do that by graphing utility versus cost as shown below:



The circle in the graph above shows the point where the slope of the curve equals 1. At this point in the curve, an additional increase in speed provides less benefit than the associated increase in cost. At any point to the left of the circle, we are "leaving money on the table" because there is a better point to the right. This is the optimal speed to specify.

Reality check

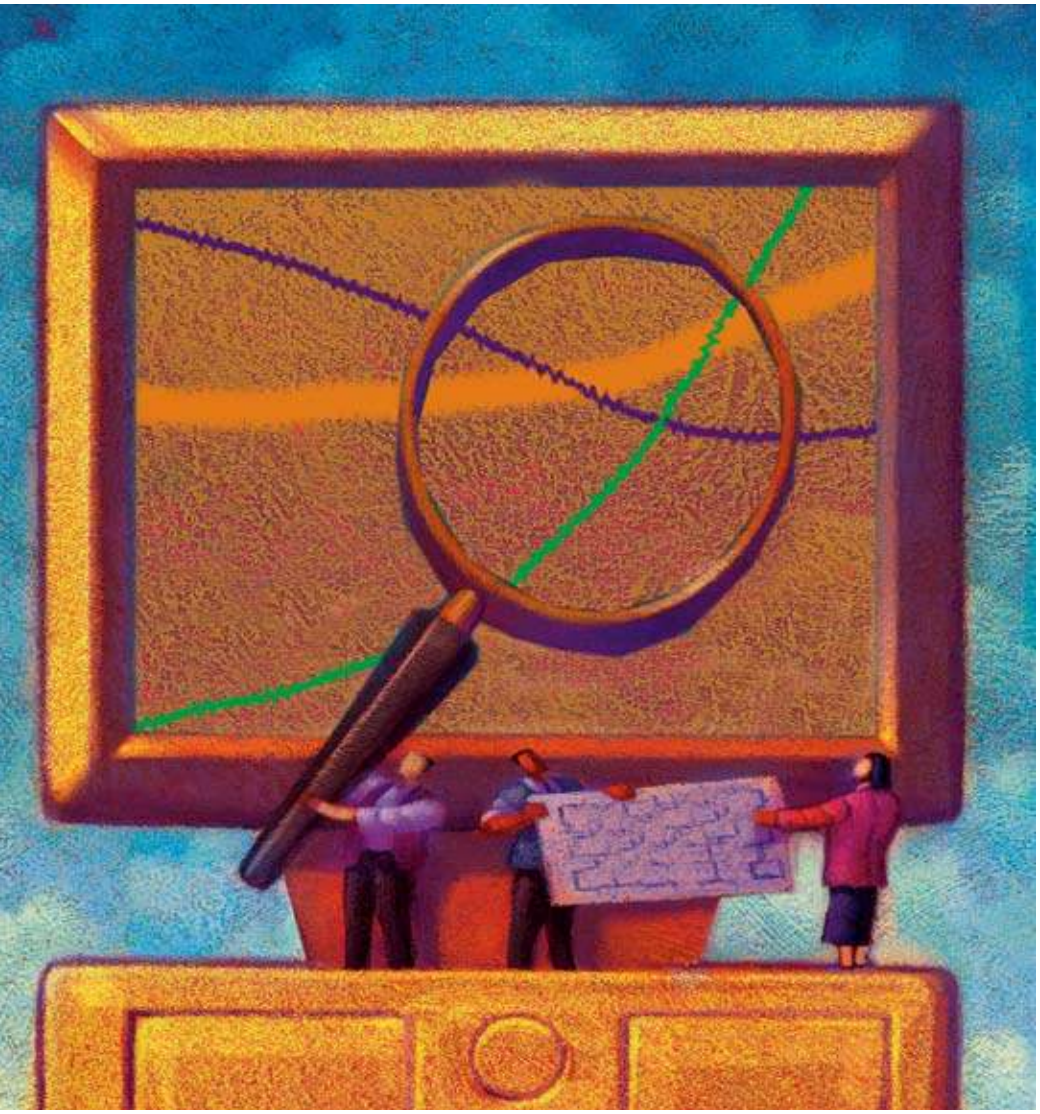
In the real world, we will not have the precise data that allows us to draw these graphs and quantitatively identify the Pareto optimal point on the cost-benefit curve. It is important to understand the fundamental principles of the trade off so that we can make informed decisions and judgment calls.

Some analyses will be relatively easy, as our development curves are usually discrete data points based on estimates of the work required to implement particular designs. We also won't have access to the full spectrum of design choices, because we will be limited by other constraints on the system as well as the creativity and capabilities of our development team in proposing alternatives.

Must be requirements

Must be requirements are the easiest to elicit and are the ones that most people consider when they talk about requirements.

Stakeholders can usually tell us what they *must* have in the software. In the *Am I hot or not?* post on requirements prioritization at *http://tynerblain.com/blog/2006/02/17/prioritizing-software-requirements-am-i-hot-or-not/,* the company 37signals focuses on this as its primary criterion for inclusion in a software initial release. They choose to only put essential, or *must be* requirements into the initial release of software.



Better not be requirements

This is just the opposite of *surprise and delight*. If dreamers think about what makes something great, then critics complain about what holds it back. This bucket does not have a place in Kano's analysis. Saying, "Users don't like confusing navigation," does not provide any benefit relative to saying, "Users prefer intuitive navigation." We suggest not using this category at all.

Apply the Kano techniques to make good prioritization decisions

1. Are our 1.0 release requirements all *must be* requirements?
2. When we specify *more is better* requirements, are they unambiguous, and are they optimal—or at least practical?
3. Do we have any *surprise and delight* requirements that will enable us to create an innovative product?

Using Kano analysis to prioritize requirements

The first release of the software should primarily include *must be* requirements.

We talked about how 37signals and other companies have taken the "more is less" approach to releasing their software. The first releases (or beta releases, which has become the misnomer du jour) of successful products have focused the majority of their efforts towards achieving these highest priority requirements.

This trend has evolved because of what used to be called "Internet time." Products are being released more quickly by startups, skunk-works, and other teams operating in Agile development environments. The dot-bomb generation of software developers has almost a decade of experience now, and is increasingly influencing company decision making, with the benefits of lessons learned about hype and substance. These teams and leaders are driving "relevant and differentiated" innovation into the marketplace faster than ever.

Geoffrey Moore, author of *Crossing the Chasm, Inside the Tornado,* and now *Dealing with Darwin: How Great Companies Innovate at Every Phase of Their Evolution,* recently posted an article *Top 10 Innovation Myths* (http://sandhill.com/opinion/editorial.php?id=66), that is in line with his new book and definitely worth a read. Mr. Moore points out that innovation is not the goal—product differentiation resulting from innovation is the goal. He is absolutely right. An innovative way to minimize the window of an application is not likely to differentiate the product from its competitors. An innovative way to automatically validate requirements would be the proverbial better mouse trap.

Making things even more competitive is the increased speed and reduced cost of getting the message out. Idea virus is a term coined by Seth Godin, and every day another company learns how to do it. When a new product gets digg'ed or slashdotted, the stampede on the server is like the mobs that sought out super cheap computers during the last holiday season. The Internet traffic can become overwhelming and shut down the servers. But that still won't stop the flow of traffic for really hot ideas—people will post links to a cached version of the page at Google™, and the idea virus keeps spreading.

It is not enough to be fast. But it is also not enough to be differentiated. Both are important—neither is sufficient alone. The twin-dynamic of smarter, faster competition combined with cheaper, faster effective marketing, demands that we focus on reducing time to market.

All of the *must be* requirements need to be included in the first release. If we could release the software without implementing features to support a *must be* requirement, then either no one will use the software, or the requirement is not a *must be* requirement.

Define and prioritize *more is better* requirements based on ROI

We have already discussed how to find the Pareto optimal point for specifying a *more is better* requirement. This is the point where additional investments in the measured characteristic are not offset by comparable gains, due to the law of diminishing returns. We showed the optimal point to be where the slope of the cost-benefit curve is 1 (or 100%). What we have not accounted for is the opportunity cost of spending those development resources on other features, capabilities, or projects. If we have a hurdle rate of 20% for investments, we should find the point on the cost-benefit curve where the slope is 1/1.2 (120% benefit for 100% cost). This normalizes our cost benefit decisions across projects.

The key to scheduling *more is better* requirements is to take advantage of the fact that they represent a continuum of performance and a continuum of benefit from that performance. Include a minimal amount of the requirement in the earliest release(s)—not the optimal amount. The optimal amount can be added later. We refer to this as requirement *staging*—or implementing portions of (or versions of) a particular requirement across multiple releases.

Surprise and delight requirements as differentiators

We care about *surprise and delight* features not because they are whimsical, but because they are valuable. A delightful splash screen does not make software easier to use, but it has value in both branding and word-of-mouth marketing. It is the buzz-marketing equivalent of great packaging for a physical product (like the ketchup bottle that stands upside down). Self-configuring software, error messages with *"click here to automatically fix it"* buttons are other examples of surprises with value. These examples can increase the number of possible users.

When prioritizing *surprise and delight* requirements, we must consider our marketing strategy to determine the importance of the requirement. Are we a startup and is this our first product? Are we a behemoth rolling out another bolt-on package for our installed base? These types of features are most valuable when individuals are making purchasing decisions and when we are relying on word-of-mouth marketing. The features are least valuable when the decision maker is three layers (and six figures) removed from the actual users. This decision maker can pragmatically decide that usability is irrelevant because he does not personally have to use it.

Summary

Kano analysis provides a powerful and easy-to-use means to classify requirements. We can use that classification to drive our prioritization decisions, making sure that we deliver all of the *must be* requirements in the first release. Kano also helps us focus on the requirements that will *differentiate* our software, while helping us make informed cost-benefit analyses on the *more is better* features.

Scott Sehlhorst has spent the last nine years providing services in enterprise software. He has done product, program and project management, consulting, development and technical presales work. He has led global teams in developing successful software and spent seven years as a design engineer. Scott formed Tyner Blain LLC in 2005 to provide targeted services for clients while building a community focused on successful software development. Contact Scott at scott@tynerblain.com, or join us on the Tyner Blain blog at http://tynerblain.com/blog

[Is your approach to product management outdated?]

think with the grid.™

Pragmatic Marketing has always been focused on the unique challenges of marketing technology products and services. The framework we teach—refined and perfected over 20 years—highlights both "best practices" and practical ones to find and develop profitable opportunities, plan market-driven products, and create winning promotional and sales campaigns. Each seminar offers immediate actionable ideas, templates, and tools.

Visit www.PragmaticMarketing.com to learn more.

PRAGMATIC MARKETING®

The Industry Standard in Technology Product Management and Marketing Education.

Thomson Medstat Transforms Product Management Through Intense Focus on Customer Needs

As part of the world's largest professional information services provider, Thomson Medstat delivers mission-critical healthcare information and software tools that help its customers better manage the cost and quality of healthcare. This requires ongoing innovation, integration, and an intimate understanding of customers' day-to-day work flow and information needs. The Medstat product management department has transformed the way it addresses these critical factors—resulting in significant improvements in R&D operations, time to market, and customer satisfaction.

Medstat is part of The Thomson Corporation, which generated $8.7 billion in 2005 by providing value-added information, software tools, and applications to more than 20 million users in the fields of law, tax, accounting, financial services, education, scientific research, and healthcare. Medstat is the leading provider of healthcare decision-support solutions. Its customers include Fortune 500 companies, more than 1,000 hospitals, state and federal government agencies, most major U.S. drug companies, and 100-plus health plans.

With a market so broad and deep, it's a monumental challenge to keep information resources on the cutting edge. That's where the Medstat Product Management team comes in. Doug Schneider, Vice President of Product Management, heads up a team of nearly 20 people charged with managing decision-support systems for Medstat. When he took over the product management helm several years ago, Schneider realized that extending the Medstat leadership position would require substantive changes—particularly in building a broader and deeper team to focus on customers' needs.

So what exactly does Product Management do?

In the past, the answer to that question was somewhat murky. "If a task didn't have a home, it landed in Product Management's lap—primarily because no one really understood Product Management's role," Schneider says. "The team used to be called Product Planning, and we were focused heavily on specifying the features and functions we were going to build for the next release. If you are always planning for the next release, you lack focus on new market innovation and you wind up chasing the market."

To help clarify the Product Management team's mission and broaden its scope, Medstat turned to Pragmatic Marketing® for insights and best practices. "The Practical Product Management® course broadened our view of product management," says Schneider. "When we looked at the Pragmatic Marketing Framework, we realized there were several areas we weren't paying enough attention to. To become a mature and effective product management organization, we needed more complete coverage across all of the boxes included in the framework.

"The other 'aha' moment in the training," Schneider continues, "was that we needed to get people into the field spending more time understanding customer needs. Too often, our product managers were getting pulled into working with Product Development on the next release and providing tactical sales support that could be done by others in the organization. That had to change."

"The other 'aha' moment in the training," Schneider continues, "was that we needed to get people into the field spending more time understanding customer needs.

Mark Gillespie, a Director of Product Management at Medstat, echoes that sentiment. "In the training, the instructor drove home the message that you're not going to learn anything sitting in your office. You have to get out there and talk to people—not just customers, but prospects. What transformed my role was becoming a market-facing product manager, getting out there and having my finger on the pulse. Yes, we need to understand what our competitors are doing. But if all we do is follow them, we're going to lose. We need to be out there ahead of them and talking to the people who are going to buy." →

Clarity and focus boost customer satisfaction and the bottom line

After going through the course, Schneider reorganized his team into three main groups. The first is market-facing product managers, who serve the four decision-support markets that Medstat supports. The second group comprises platform-facing product managers, who oversee the company's software assets. The third is a senior product manager responsible for driving new market innovation.

"Implementing market-facing product managers put a lot of clarity into our organization," says Schneider. "Now there is a single focal point in Product Management for each market. More importantly, our product managers now spend more time in the field and less time looking over Product Development's shoulder."

The new organizational structure wasn't the only transformation. Medstat also revamped its processes for defining and developing product requirements. "In the past, we had a traditional waterfall approach to the requirements process," Gillespie recalls. "We would gather some information from customers, write a requirements document, and then throw it over the wall to Software Development. The problem was, we would get to the end of a software release and find that there was a gap between what Product Management was expecting and what Software Development delivered.

"Now, before we put pen to paper," he says, "we conduct brainstorming sessions with the development team. We articulate where we want to go, and they can ask questions and try out ideas for solutions. As a result, the development team has a better understanding of the use of the product in the market. That front-end process has helped us close the gap between the vision and the reality of what's delivered."

The impact of the transformation has been substantial—both on customer satisfaction and the bottom line. "What we're really going for in Product Management is a good return on investment for our R&D money," Schneider explains. "In the last few years, we have significantly improved the amount of revenue we are generating per dollar of R&D. In other words, we have improved our innovation batting average by better understanding market needs. That also translates into a significant improvement in our customer satisfaction measurements—a 15% jump over the last three years across a very large customer base."

Gillespie provides further evidence. "In the last year, we have seen a noticeable difference in the feedback from our Client Advisory Council. They are saying, 'You are really hitting the mark and delivering exactly what we told you we wanted.' That is directly attributable to the improvements in our requirements processes and in our new market-facing attitude."

Solution rollouts vs. feature rollouts

Another important transformation occurred in the product rollout process at Medstat. "For a long time, we recognized the need to improve our go-to-market methodology, but the Pragmatic Marketing training gave us a standard framework for doing that effectively—including articulating strong value propositions and win themes," Schneider explains.

Says Gillespie: "Previously, we would schedule meetings with marketing and sales teams a month before a release to demo product functionality and review a bullet list of features. Since going through the Pragmatic Marketing training, we are focused on the business value. Now we conduct 'solution rollouts,' in which we talk about what's happening in the market, the latest buzz, what the competition is doing, our 'marketecture,' and the message we want to communicate to specific constituents. That has been very successful with the salespeople. And we are able to do that three to six months before a product is due to release. This gives Sales more lead time to talk with customers about what's coming up and to include that positioning in RFPs."

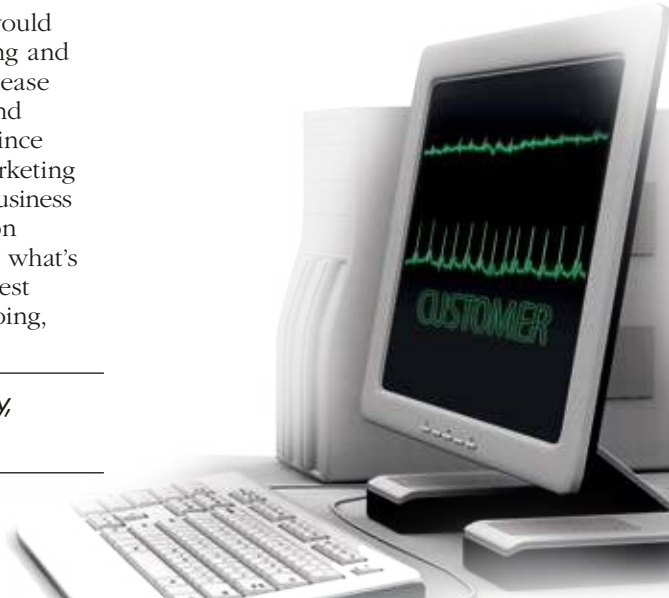
The impact on time-to-market has been profound. "Before, it took us 12 to 18 months to absorb a new capability into the field organization and start selling it effectively. We would have great new product capabilities, but it took a long time for the market to understand we had them. Now that is happening within 60 days of the product rollout."

Boosting the "confidence factor"

By listening to the market, refining the requirements process, and strengthening the rollout process, Medstat also has boosted its sales and marketing teams' confidence in communicating new innovations to the market.

"Today, by the time we baseline the requirements, we have a 90% confidence level in what will come out of the gates," Gillespie says. "Our sales and marketing teams now have the right kind of information—well ahead of the curve—to talk to prospects with confidence and to lead the market. And leading the market is exactly where Medstat wants to remain." pm.c

THOMSON MEDSTAT



To feature the product management success at your company, contact info@PragmaticMarketing.com

2nd ANNUAL

June 13-14, 2006 – Santa Clara Convention Center

2006 SOFTWARE Marketing Perspectives

CONFERENCE & EXPO

The Insider's Secrets to Success

When was the last time you were able to compare notes and learn from the top software product managers and marketers in a collaborative environment? The second Software Marketing Perspectives event in Santa Clara, CA offers a unique opportunity to "drink from the fire hose" and gain more insight in a two-day forum with top experts in software product management, software marketing communications, public relations and business development.

Are you a high technology software product management or marketing professional? Would you rather take the elevator to top performance with your peers rather than climb the mountain alone? Come to the Software Marketing Perspectives event to learn proven strategies from your peers that enable you to:

- Negotiate priorities and build better products
- Build improved online relationships with clients
- Grow revenue by anticipating market trends and change

If you are wondering how it is possible to increase revenue and improve marketing performance, then come to the SMP event, where your peers will share the Insider's Secrets to Success. We learned in October 2005 at the first SMP Event in Cambridge, MA that we all struggle with the same problems. Yet some of our peers have found solutions that work. Come to the SMP event and get the inside scoop on what works and what doesn't from your peers — product managers, marketing managers and professionals engaged in the business of software marketing. Learn from those who do it now.

Conference Program Overview

- Product Management
- Marketing Communications and Public Relations
- New Models: Software As A Service

2006 EXHIBITORS

Pragmatic Marketing

Ryma Technology Solutions, developers of Feature Plan™

Autodemo®

LEVERAGE SOFTWARE

LCG

O'Keeffe & Company

OpSource™ The SaaS Experts

pathfinder

2006 MARKETING PARTNERS

SVPMA Silicon Valley Product Management Association — Silicon Valley

ukpmf — UK

TPMA — Toronto

VPMA — Vancouver

BPMA Boston Product Management Association — Boston

To get more information and to register, visit:

www.smpevent.com

Upcoming Pragmatic Marketing Seminars

Call (800) 816-7861 or go to www.PragmaticMarketing.com to register!

Practical Product Management®

Introduces a framework that gives product managers the tools to deliver market-driven products that people want to buy. Focuses on the practical aspects of juggling daily tactical demands of supporting the channel with strategic activities necessary to become expert on the market.

June 5 - 6 (7)*	Chicago, IL
June 5 - 6 (7)*	San Francisco, CA
June 20 - 21 (22)*	Toronto, ON, Canada
June 26 - 27 (28)*	Bedford, MA
June 26 - 27 (28)*	Santa Clara, CA
July 10 - 11 (12)*	Vancouver, BC Canada
July 17 - 18 (19)*	San Francisco, CA
July 24 - 25 (26)*	Austin, TX
July 31 - August 1 (2)*	Burlington, MA
August 7 - 8 (9)*	San Jose, CA
August 14 - 15 (16)*	Pittsburgh, PA
August 28 - 29 (30)*	San Francisco, CA
September 11 - 12 (13)*	Bedford, MA
September 18 - 19 (20)*	San Francisco, CA
October 23 - 24 (25)*	Toronto, ON, Canada
December 11 - 12 (13)*	Boston, MA

*Day 3 is Requirements That Work

Requirements That Work™

Provides a repeatable method for product planning resulting in a Market Requirements Document that others read and use. Establishes clear roles for product planning team members and teaches a process that creates an executable plan that delivers solutions that sell.

June 7	Chicago, IL
June 7	San Francisco, CA
June 22	Toronto, ON, Canada
June 28	Bedford, MA
June 28	Santa Clara, CA
July 12	Vancouver, BC Canada
July 19	San Francisco, CA
July 26	Austin, TX
August 2	Burlington, MA
August 9	San Jose, CA
August 16	Pittsburgh, PA
August 30	San Francisco, CA
September 13	Bedford, MA
September 20	San Francisco, CA
October 25	Toronto, ON, Canada
December 13	Boston, MA

Effective Marketing Programs™

Delivers practical tools and processes for product marketing, industry marketing and marketing communications managers who want to improve their strategic contribution and align with the sales organizations. Learn how to develop, build and measure programs that are more than a revised list of last year's tactics.

June 7 - 8	San Francisco, CA
July 26 - 27	Boston, MA
August 30 - 31	San Francisco, CA
September 13 - 14	Boston, MA

Not sure Practical Product Management is for you?
Attend this **FREE** 1/2-day session.

The Strategic Role of Product Management™

Registration for this **FREE** seminar is required via our website: www.PragmaticMarketing.com

A subset of the two-day Practical Product Management seminar, this session introduces the industry standard for technology marketing. Shows how Product Management and Marketing personnel can move from tactical to strategic activities.

June 8	Chicago, IL
July 13	Vancouver, BC Canada
July 27	Austin, TX
August 10	San Jose, CA
October 26	Toronto, ON, Canada

New seminar locations!

Here is your opportunity to **'Live the Grid!'** with Pragmatic Marketing merchandise.

Go to www.PragmaticMarketing.com/shop for shirts, posters, coffee mugs, and more.

productmarketing.com™
16035 N. 80th Street, Suite F
Scottsdale, AZ 85260

PRSRT STD
US POSTAGE
PAID
PHOENIX, AZ
PERMIT #995