October Event with Sridhar Ramanathan, President of the Pacifica Group
Is Your Product "Channel Friendly"?
By Greg Cohen

SVPMA was excited to have Sridhar Ramanathan, President of the Pacifica Group, return to address the association at its October 4, 2006 meeting on making your product channel friendly. The speaker presented a broad overview of the channel: what it is, what it does, and what makes it tick. He then challenged the audience with an eight point test to gauge their products' "channel friendliness."

Defining the channel depends on your perspective. Vendors and customers view the channel differently.

The vendor views the channel as a route to market for its product. The customer views it as the place where they buy and get help on products. The speaker discussed the Toyota car he purchased as an example. The dealer showed him the car, negotiated the price, took the order, handled financing, had the car in inventory, and now assists in maintaining and servicing the vehicle. Although Sridhar purchased a Toyota, he has never once dealt directly with the manufacturer.

There are many types of channels: direct sales over the internet, telemarketing, retail stores, distributors, value-added partners, and a direct sales force. Each type has a different level of value-add and transaction cost. The remainder of the presentation focused on issues relating to

Sponsors

Feature Plan

VISTAAR®

ACEPT

Inside this issue

----

and More

How do I decide how long to allow for a localization cycle?

By Andrew Jones, Senior Project Manager, Rubric

Vendors offer faster, smarter turnarounds and translation memory tools are designed to help speed up the localization process. But how long does it really take to localize a product?

Your product is selling well in the marketplace. Your marketing team is receiving requests for versions in other languages. They ask you how long it will take before you can give your customer a version in Spanish, Brazilian, Portuguese and Arabic. Can they have it by the end of the week? The end of the month?

Having never produced a different language version before, you are stumped by this question and reply that you had better contact a few localization vendors to gauge their opinion and report back on your findings. But marketing, as always, is impatient – they need to give their customer an release date estimate, and they don't have time to wait for a survey. Can't you give them an estimate now? How does the localization vendor decide how long something is going to take anyway?

Letter from the President

As we near the end of 2006 it's been a great year for SVPMA. We've had excellent monthly speakers, moved to the Hilton as our regular venue, held three social networking events and had well-attended and informative quarterly workshops. Our membership has continued to grow - we're at almost 500 paying members, and our announcement list has over 2000 email subscribers.

To celebrate our continued success, we have landed a very prominent speaker, Geoffrey Moore, the author of Crossing the Chasm, Inside the Tornado and Dealing with Darwin, so be sure to mark you calendars for December 6th. This special event will be held at the Computer History Museum in Mountain View (Shoreline and 101). Geoffrey is world-famous for his books and is an excellent speaker, and the Computer History Museum is a fascinating venue. We expect this to be the largest SVPMA event ever, so make sure you don't miss it!

Also worth noting: SVPMA's quarterly workshop is coming up on November 18th with Patti Wilson of The Career Company. Patty will be talking about Influence Network-

ing: A Strategic Roadmap to Change Careers or Penetrate a Business Sector. I've been to several of Patti's classes before and found them to be VERY useful, so take advantage of this great opportunity.

Wishing you all a great 2006 holiday season!

Brian Lawley

President, SVPMA

SVPMA is an all volunteer non-profit association for Product Managers in Silicon Valley and around the San Francisco Bay area.

Our mission is:
- To provide a forum to share day-to-day experiences and insights in Product Management
- To create a safe network of peers
- To promote research and education in Product Management

Board
Brian Lawley…. . . . . President
Gregory Cohen. . . . . Director
Sachin Gangupantula . . . . Bus Dev & Website
Mike Freier. . . . . Membership
Jamie O'Halloran. . . . . Workshops
Neeta Godhwani . . . . Marketing
Ray Stendall . . . . Finance
Sharon Grimshaw . . . . Volunteers

Managing Editor
Sarah Lawley

Contributing Editors

Gregory Cohen - 280 Group
Brian Lawley - 280 Group
Andrew Jones - Rubric
Srini Namineni - IBM

To learn more about the SVPMA go to
http://www.svpma.org
Or email to newsletter@svpma.org

Mark your Calendars!
Upcoming SVPMA speaker AND other events :

- December 6th, 2006 - **Geoffrey Moore @ The Computer History Museum**
- November 18th, 2006 - **Quarterly Workshop - Patti Wilson - Career Company @ Hilton Santa Clara**
- January 10th, 2007 - **Ken Norton - JotSpot**
- Quarterly Networking Event– Details coming soon

For more information please go to www.svpma.org

Enroll or **Renew** your SVPMA membership now for an incredible annual membership price of only $75.

Come see leading speakers from the Valley's most innovative and respected companies share their secrets of product management and marketing success.

Membership benefits include:

- Free priority entry to SVPMA monthly events! ($240 value)
- Access to the LinkedIn SVPMA group with one-degree access to other product management professionals
- Access to Product Management hiring managers and recruiters using SVPMA
- Networking opportunities with top-tier PM talent
- Education opportunities with industry experts to learn best practices

---

Thanks to these great SVPMA volunteers!

* **Event Registration: September/October**
  * Brian Sohmers
  * Kameliya Vladimirova

* **Event Photography**
  * Subbu Swaminathan (a) Shankar
  * Corina Mitu

* **Event Writer**
  * Srini Namineni

~

Interested in volunteering?
Contact Sharon Grimshaw, Director of Volunteers
sharon@svpma.org

Brian Lawley, President and Founder of 280 Group LLC, spoke about the various types of roadmaps and an eight step process to create product roadmaps.

He started by talking about the variety of situations where roadmaps are needed. Sales teams may need a roadmap to reassure prospects and customers about a product's future direction. Development teams like to have a roadmap to guide their engineering efforts and resource allocation. A startup company may have to show a roadmap to get its next round of funding. Executives need roadmaps to brief press or analysts on where the company is headed.

Brain mentioned five types of most commonly used roadmaps: Market & Strategy, Visionary, Technology, Platform and Product (Internal & External). Each type of roadmap is good at meeting specific needs for a specific audience.

Market & Strategy roadmaps describe target market segments and high level plans to develop products for those segments. **Visionary roadmaps** paint a big picture of trends in the industry and society and then draw linkages to the products that will be offered to leverage those trends. **Technology roadmaps** are similar to the visionary roadmaps except that they focus on technology trends.

Platform roadmaps show plans for core platform, products that will be built on that platform as well as the associated development tools. These platform roadmaps are targeted at partners who develop products on the platform.

Product roadmaps show what product releases will be available when and the salient features of each product release. External product roadmaps are targeted towards customers, partners, press and analysts. Internal product roadmaps are meant for internal communication and usually contain more specific details than corresponding external roadmaps.

Brian then went on to describe the following eight step process to developing product roadmaps.

1. **Decide the detail level & amount of time you want to spend**: You need to decide up front whether you want a quick & dirty process or a very rigorous process in creating the roadmap. According to Brian, most companies tend to land somewhere in between these two extremes.

2. **Assess competitive moves, market & technology trends**: In this step, you need to assess the market landscape. For example, is the market in an early adopter stage or is it in a widespread adoption stage? What are the important verticals? What are the technology trends?

3. **Gather & Prioritize requirements**: Get requirements from all sources including engineering, customer support, executive pet features etc… and create a master list of requirements in a spreadsheet. Create criteria columns such as pain for the user, percentage of customers impacted, up selling existing customers and revenue from new customers etc… In this feature prioritization matrix spreadsheet, score each requirement against each criterion. Choose weights for the criteria and calculate a weighted score for each requirement. Use this score to rank the requirements.

4. **Decide on the time frame covered by roadmap**: Do you want to create a roadmap for the next three months or the next ten years? You can also do a combination such as providing quarterly roadmap for the next four quarters and a yearly roadmap for subsequent years.

5. **Choose an organizing strategy for features**: You can choose to organize the product releases by themes (set of features addressing a particular area), a golden feature (release will not happen without that feature) or a timed release strategy (train model). Choose a strategy that works well for your product after carefully weighing the pros and cons.
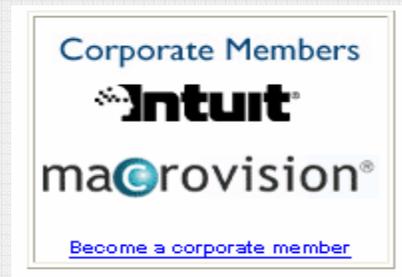
Creating Compelling Product Roadmaps part 3

(continued from the September/October SVPMA newsletter)
By Brian Lawley

In the last issue we discussed an eight step process for creating product roadmaps and how to prioritize requirements. Now we'll talk about how to logically organize the features once you have them prioritized.

Strategies for Organizing Features
Once you have gathered and prioritized the features, you need to decide which strategy you want to use to organize them. There are three strategies generally employed: Themes, Golden Features and Timed Releases.

Themes
Themes are a simple and effective way to organize your releases. To create a theme take the original list of prioritized feature requests and begin to classify them into similar categories to see if any trends emerge. For example, it might be that the majority of requests are for product stability or for increased performance. There might be a trend towards better security in the product, or a trend towards multi-user collaboration. The key is to identify whether there are one or two high-level categories that have a large number of features that fall logically under them. You can then group them into releases to put on the roadmap, each with its own theme.

Golden Feature
The idea behind the Golden Feature technique is that you choose a single feature that is the absolute most important priority for the release, and then you get the entire team to rally around it as the focus. This single feature provides enough customer value to hang the entire release on. It is compelling enough that all of your customers will want to upgrade (and some potential new customers will be further convinced to purchase). And it is simple enough to communicate from a marketing and competitive point of view that it gives your product a noticeable boost. Ideally you should be able to describe the Golden Feature in one to five words. Some examples might include: 1.) 30% Faster Performance 2.) Import MS Word files or 3.) Double your battery life. It has to be simple and very compelling - if your release has a long list of features, but you can't find a one to five word value message to communicate, it's going to be very hard to build a marketing campaign and keep your product focus.

The Golden Feature technique is particularly effective if you are on short release cycles or are doing a point release (1.1, 1.2, etc).

Many of your team members may want to include all kinds of other features that can be of varying interest and value to your customers.

If you can get them focused on the one Golden Feature for the release (and in agreement that everything else is a "nice to have" priority but that you won't hold up the release for it), you'll have a much better chance of success. You'll also have a higher likelihood of meeting a tight schedule, as you can plan the release with more certainty if the Golden Feature is the only critical path item.

Timed Release
The timed release strategy uses pre-determined release dates to drive the schedule rather than features. You decide on a release interval (quarterly, every six months, once a year, etc.) and then you take the prioritized feature list and estimate which of the top features can make it into each release. If a feature misses the release it gets moved to the next release.

The timed release strategy has the advantage of providing your customers, partners and internal company constituents with known ship dates they can plan around. It also alleviates the pressure of everyone trying to vie to get their "pet" feature into the next release, since they know that there will be another release coming along shortly. The downside of the timed release strategy is that it doesn't work very well for features that have long development cycles.

If your team is working on a feature that will take several release cycles to complete they may be pulled off to finish work for the interim releases.

In the next issue we'll talk about how to create roadmaps for multiple product lines and include some additional best practices tips. ⌘

Brian Lawley is the President of SVPMA and the 280 Group, which provides Marketing and Product Management Consulting and Contractors to help companies define, launch and market breakthrough new products. The 280 Group sells the Product Roadmap Toolkit, which includes a narrated seminar, 75 roadmap templates and prioritization matrices .
For more information go to www.280group.com.

Nailing down a localization schedule is as painful as managing your boss, your production engineers, and your marketing department. So I'm here to help you by illustrating some of the factors you must consider, and set you thinking through the problem. Drawing up a localization schedule is not straightforward given the many aspects you must take into account – and not just the customer's desired release-to-market date. Unfortunately, there is no magic formula to working out how many days a localization schedule will take – if there were, this would be a much shorter article!

What you might be thinking now is, "Why doesn't he just give me a table of metrics, such has how many pages can be DTPd in a day, or how many words can be translated"? What I want to do is to show that each stage in the localization cycle needs to be broken down and fully understood before a realistic schedule can be established.

Metrics alone are misleading. Let me explain the limitations of metrics, and then look at how to break down the cycle.

> *First of all, let's make sure that we haven't forgotten the basic premise of this article, which is to help you <u>estimate</u> how long the cycle is going to take. You can leave the really precise calculations to the vendor: what you want to avoid is telling your manager that something can be done in six hours when it is going to take six weeks.*

There are three main problems with metrics.

Averages aren't: There is a lot more to each individual task in the localization process than is apparent at first sight. For example you may hear that an individual translator can translate about 2500 words a day. This is a fairly standard metric – of course, you will always find someone who will tell you it is less or more than this, but this seems a reasonable basic figure. So if you have a documentation file with 5000 words to translate, it can be done in two days, right? Well . . .

- Have you considered that it is not just a question of the translator sitting down immediately and typing 2500 words?

- Have you considered the time taken for the translator to research the material being translated, to assemble glossaries and reference material which is essential before starting?

- Have you considered that the file may need to be prepped for translation so that it is in a format that the translators can work with?

- Have you considered that the translation needs to be proofread and edited as well?

Have you considered that quality translators are always based in the country in question and therefore there are time zones to be considered, and that the translator might not be working the same hours as you are?

As you can see, the translation task alone (we haven't even considered layout and graphic work that may or may not be needed as well) is actually a set of tasks with interdependencies, and here the metric alone doesn't really help in planning. Instead, it is better to break down the task into all its component parts, and then afterwards apply the metrics that you know. Of course, 100,000 words are going to take longer than 100 words, but the number of words is not the only factor.

Resources are always limited: A second problem with metrics is that they are effectively meaningless unless you have a control over the number of resources that are going to be used. For example, with your 5000 words of documentation, ten translators are going to complete it in much less time than one translator. However, the likelihood is that your localization vendor is going to be controlling the number of people that are going to do the translation, not you: encouraging them to throw resources at the job to speed up the process is not really a wise move anyway – ten translators working on 5000 words is going to lead to much worse quality than one translator working on it, as there are much greater risks of inconsistency.

Moreover, let's look back at the "sub-tasks" to translation themselves.

- Can assembling glossaries and reference material be speeded up by using multiple resources? Not really.

- Can prepping the file for translation be speeded up by dividing it into ten and having ten resources each prep their little section and then stick it back together? Not really.

If you come to the vendor with a requested schedule in which you have already factored in the sub-tasks that make up the translation task, then not only will you feel in control of the schedule from the outset, but you will not be giving the vendor the opportunity to risk quality at the expense of speed.

Incomplete inventory: Thirdly, when you are at this stage of estimating your schedule, you might not even know how many words, pages, or screens your product has, and that might need a complex analysis process to ascertain. That being the case, knowing how many words a translator, or several translators, can translate in a day is going to be of no use to you at all. Yet you still want to have a basic idea.

There must be a better way!
A better way to sketch your basic schedule is to break it down

into tasks and sub-tasks. Herein, I'm going to consider an entire software product. Broadly speaking, a localization cycle for software needs to include the following:

1. Planning stage including budgeting and selection of resources

2. Initial testing of suitability for localization

3. Preparation of the software for translation

4. Translation and proofreading of the software

5. Review of the translation

6. Building of the software to include the translations

7. Testing of the localized software

8. Bug fixing the localized software

9. Integration of the localized software

10. Client acceptance of the software

Each of these ten stages in itself is a complex set of sub-tasks with interdependencies, and each of them will depend on the size of the software involved. But a good way to come up with the interdependencies is to ask yourself a list

of questions about each of the stages. This is likely to lead to a resultant list of tasks.

Let's consider just some of these tasks and sub-tasks here: once you can see how I am breaking down three stages into some questions, I hope you will be able to do the same with the other sections in order to come up with an idea of how long you need.

Stage 1) Planning Stage

This could potentially be the most time consuming stage of the whole process, and if you are not careful you could lose weeks before a single word is even translated. Here are some things you need to consider in this stage:

- How many people are going to be involved in selecting the resources?

- Is it going to be entirely outsourced or will some of the localization work (e.g., testing) be done in-house?

- Do you need to get internal budgetary approval to go ahead with the localization and how long will that take?

- Do you need to get external approval (e.g., from your own clients)?

Stage 5) Review of the Translation

- Where is the review going to take place?

- How many review cycles do you need?

- How many different people need to review it?

- Does the software need to be built before it can be reviewed?

Stage 6) Testing of the Software

- Do you have a testscript or does one need to be written?

- Do you want the vendor to do a full test or just a smoke test?

- How many platforms do you need to test on?

- How many internal groups will want to test the software?

- Have you factored in a regression test once bugs are fixed?

Within each of the stages, asking the questions above will probably lead to a task on your schedule. If you come up with five questions (and therefore tasks for each stage) you are going to have 50 tasks all in all. I would recommend that you

schedule each of these tasks to take a day (although you know that an individual task might not actually take that long, some will take considerably longer, so one day is a good average as a basis). That means a you will have a basic schedule of around 50 working days.

Before you fall off your chair in horror because your software only contains 100 words, don't panic just yet, because some tasks can be done in parallel, and you will find that this will reduce your basic schedule length significantly. Moreover, if your software really is only 100 words you are likely to have a lot fewer questions to ask about it, and consequently fewer tasks. Conversely, if your software is huge, you will doubtless find more questions and tasks.

Once you have your basic schedule, then you can start to add a few or a lot of days to some of the tasks depending on how large you think your product is. This might sound like pure guesswork, but in terms of an estimated cycle, the number of days you allow for each task is less critical than making sure you have the right number of tasks in place.

The list of tasks, which I have demonstrated above, can be assembled by stepping back and asking yourself some simple questions. But these questions are the critical element in drawing up a basic localization schedule. ⌘

Rubric assures a better localization experience. Rubric's flexibility, on-demand scalability, and integrity, guarantees localization success. Through planning, anticipating, responding – Rubric's proven project management methods get the job done, and accelerate your products into international markets. Rubric works collaboratively with you to meet your changing needs with excellent execution. This is why [Rubric scores a 99% customer satisfaction rating](). www.Rubric.com

6. **Build internal roadmap**: The internal roadmap must have specific details of primary and secondary features in each product release and a credible timeline. Compare this roadmap against competitor's roadmaps as a cross check.

7. **Fine tune and get buy-in**: Socialize the internal roadmap with the engineering, sales, tech support teams and management. Fine tune by incorporating their feedback and achieve buy-in from all the stakeholders.

8. **Create external roadmap based on internal roadmap:** To minimize the damage from slipping a product release, be as vague as you can get away with. For example: specify high level themes instead of individual features. Don't put specific dates. Use code names instead of version numbers.

Brian also talked about creating roadmaps for multiple product lines. Create a roadmap for each product line by following the eight step process described earlier. You just need to go through additional steps of prioritizing product lines and creating a high level combined product line roadmap.

In Summary, product roadmaps are critical to win and keep customers, guide engineering, drive strategic planning and to communicate company's vision to partners, press and analysts. If you're responsible for developing roadmaps for your company's products, first figure out what type of roadmaps are needed. You should be able to create compelling product roadmaps for your company by methodically following the eight step process prescribed by Brian. ⌘

As a Senior Product Marketing Manager, Srini is responsible for strategy and planning of IBM's enterprise master data management software. He has over 15 years of industry experience in business intelligence, web applications and system management arenas. He can be reached at snaminen@us.ibm.com.

five types of indirect channel providers:

- OEMs – Dell, HP, IBM. OEMs are driven by Average Selling Price (ASP) or contribution margin. They ask the question if I add your component (e.g. disk drive) to my product, how much more can I charge and what will be my contribution margin?
- Value Added Reseller (VAR) – ACS, Getronics, En Pointe. VARs are driven by profit margin and volume. They do not make any products, but sell and service other vendors' products.
- System Integrators – IBM, EDS, HP, Accenture. System Integrators are driven by client value. They develop on-going relationships and measure their success on client retention and growth. Service revenue, rather than the sale of product, comprises the majority of their sale.
- Distributor – Ingram Micro, CDW, Comstor. Distributors are driven by volume and dollar margin.
- Service Providers – AT&T, BT, and Orange. Service providers are cost focused and can represent a large opportunity. SBC, for example, is one of Cisco's largest resellers.

Using an indirect channel requires product management to think beyond the feature set and focus on the whole product, including tech support, training, marketing programs, service, etc. Mr. Ramanathan then presented his eight point list to test if a product is "Channel Friendly"

- Do you have a simple and compelling, one sentence value proposition?
- Do you offer attractive margins (>30% plus more for selling support contracts and deal registration) for the channel?

- Can a sales rep prepare a quote in less than 3 minutes?
- Is the sales cycle less than 90 days?
- Do you have a killer demo (i.e. can the sales rep wow a prospect in less than 15 minutes?)
- Does your product allow your channel partner to leverage the sale to make further revenue on planning, design, implementation, technical support, and customer education (target $1 - $3 of additional revenue for every $1 of your product sold?)
- Do you create a reason for your rep to sell again (renewal, upsell, cross-sell) into an account in less than six months.
- Do you have a goal to drive down support cases through improved usability and design?

If you can answer "yes" to the eight questions above, you are in good shape for the channel. Most likely, you at least have a few areas on which to work. As product managers it is important to realize how important the channel is to your product's success, and channel considerations must be designed upfront into the product strategy. By applying the eight point test, you might even be the next VMware, which went from zero to $100M in sales in just four years through the channel. ⌘

Greg Cohen is the Director of Business Development and Principal Consultant at the 280 Group and on the board of the Silicon Valley Product Management Association. He has over ten years of product management and marketing experience with a focus on application services specifically with spend analysis, business analytics, and contract management.