Software Product Management

Dr. Christof Ebert
Vector Consulting Services

It's easy to confuse the disciplines of project manager and product manager. Simply put, the development of the product or service falls to the project manager, while the market success of software and system products depends on the skills and competence of the product manager. This article provides an overview of software product management and the role of a product manager, and describes concrete practices that can boost an organization's software product management and thus the success rate of products in terms of predictability, quality, and efficiency.

Imagine loggers in a forest. They work hard and cut tree after tree. It is a huge physical effort and their foreman drives them hard to stay on schedule. He wants to cut a certain number of trees per day and provides the workers with all they need to achieve this objective. Suddenly the client shouts, "you cut down the wrong trees!" Despite all the hard work of the foreman and his team, they did not manage to deliver the intended customer expectation. Sound familiar? Indeed, this is what I've observed with many software products. Organizations are pushed to the extreme to be ever more efficient and create products at a low cost, but when it hits the market and sales are lower than expected or customers demand several changes during the development process, margins are dramatically reduced from initial targets.

Successful product management means delivering the right products at the right time for the right markets. Naturally, the success of a product depends on many factors and stakeholders. However, it makes a big difference when a person is empowered to manage a product from inception to market and evolution—and the same person is held accountable for the results. This is the product manager.

At Vector Consulting Services, we have learned from experience with many clients in different industries that success comes from anticipating and meeting the customer's needs together with being on time and on budget. Technical product development—such as for automotive components, communication solutions, defense systems, or IT infrastructure—traditionally focuses on the project perspective and operationally executing a set of given constraints within the triangle of content, budget, and time. Often, it becomes clear too late that customer needs were different from what is built.

Project execution can be rather easily improved by means of CMMI®. Today, there are a lot of exciting results from optimizing projects in terms of cost and cycle time [1, 2]. However, the software product management responsibility and underlying processes remain vague. I often see product definition, road mapping, and marketing decoupled from the engineering project-related processes, which creates deficiencies and overheads such as heavy changes in requirements and missed market opportunities. It is like the loggers: The project runs well, but with the wrong results.

While an organization can embark on the general principles of product management [3], not much specific guidance is available for software product management. This article will provide a small introduction and tutorial on software product management.

What Is Software Product Management?

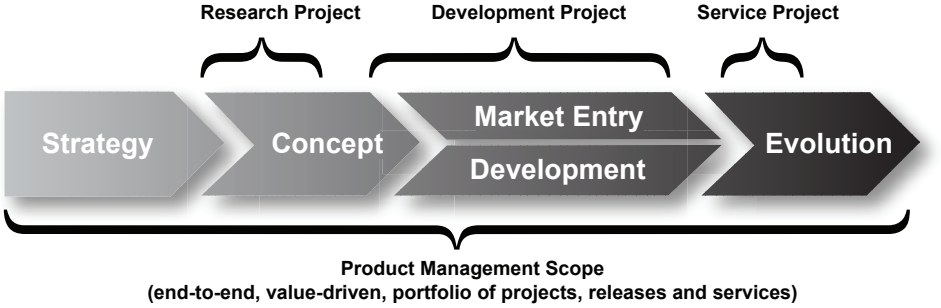
Product management is the discipline and business process governing a product from its inception to the market or customer delivery and service in order to generate the largest possible value to a business. A product is a deliverable that has a value and provides an experience to its users. It can be a combination of systems, solutions, materials, and services delivered. Product management provides leadership to activities such as portfolio management, strategy definition, product marketing, and product development.

Often, the roles of product manager, project manager, and marketing manager are unclear in their distinct responsibilities. To successfully define, engineer, produce, and deliver a product, these three roles need to be clarified [3, 4, 5]. Figure 1 provides an overview of an archetypical product life cycle and shows how different projects integrate towards an end-to-end view of the product. It highlights the differences between managing a project and managing a product. The project is a temporary endeavor undertaken to create a product. The project manager focuses on delivering one specific product or release while meeting time, budget, and quality requirements. The product manager looks to the overall market success and evolution of this product together with its subsequent releases, related services, etc.

To clearly assign responsibilities, there should be three distinct managerial roles:

- The product manager leads and manages one or several products from inception to phase-out in order to maximize business value. They work with marketing, sales, engineering, finance, quality, manufacturing, and installation to make the products a business success [3]. They have business responsibility beyond the single project. They determine what to make and how to

Figure 1: *Software Product Management Spans the Entire Product Life Cycle*



® The CMMI is registered in the U.S. Patent and Trademark Office by Carnegie Mellon University.

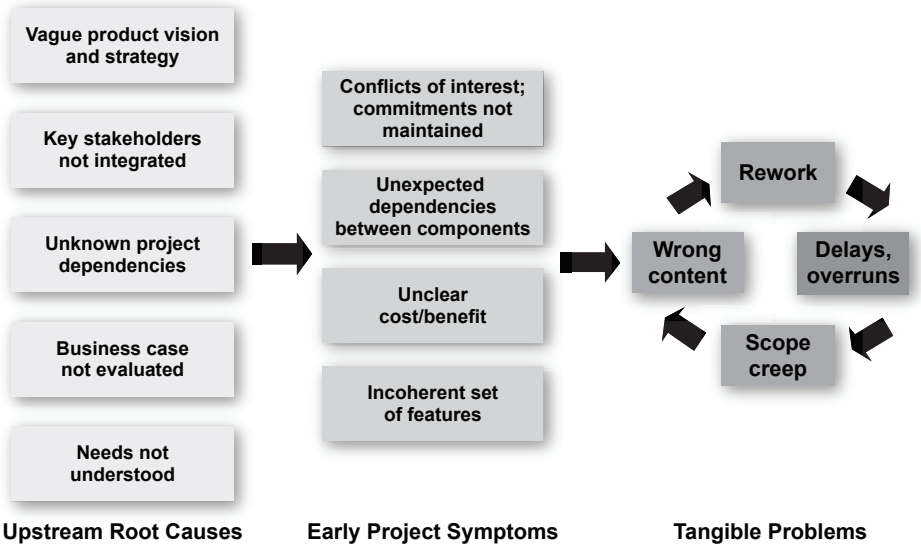


Figure 2: *The Results of Insufficient Product Management*

produce it, and are accountable for business success within an entire portfolio. They approve the roadmap and content and determine what and how to innovate, and are responsible for the entire value chain of a product following the life cycle, asking: What do we keep, what do we evolve, and what do we stop?

- The project manager determines how to best execute a project or contract. They ensure that the specific project is executed as defined and are accountable for business and customer success within a contract project. They manage the project plan and its execution and ask: How do we get all of this accomplished?
- The marketing manager determines how to sell a product or service in order to create a customer experience. They are accountable for market and customer success and have a profound understanding of customer needs, market trends, sales perspectives, and competitors. The marketing manager communicates the value proposition to sales and customers, drives the sales plan and execution, and asks: What markets will we address?

One might argue that in many organizations, one or several of these roles are laid out differently and might simply be coordinating based on directions received from management. While this has certainly been observed, such organizations often encounter interface and responsibility battles and have a lack of ownership as a result. These three roles are necessary and need to be empowered—and held accountable for results. This not only stimulates motivation, but also facilitates faster and more effective decision making in a company [2, 3].

Over the years, Vector Consulting Services has investigated root causes of such insufficient product management and its impacts on hundreds of technical products with different origins, development paces, and sizes [1, 4]. Figure 2 provides an example of how product management failures cause rework, scope creep, and delays. Insufficient product management typically lacks vision, has an unclear market and business understanding, and doesn't involve the right stakeholders (see the left side of Figure 2). This leads to initial symptoms such as a conflict of interest on priorities and contents and incomplete requirements. From here, it's a vicious circle with changes that necessitate rework, which in turn causes delays, which in turn causes scope creep—and so on. Poor product management causes insufficient project planning, continuous changes in the requirements and project scope, configuration problems, and defects. The obvious (yet late) symptoms are more delays and overall customer dissatisfaction due to not keeping commitments or not getting the product they expect (the right side of Figure 2). Being late with a product in its market has immediate and tremendous business impacts [6, 7, 8]. In the contract business, this often means penalties and, in practically all markets, it reduces customer loyalty and overall sales returns.

The tangible problems can't be fixed by pushing a button; instead, the upstream root causes need to be fixed. It would be fatalistic to just take it for granted that requirements changes will always cause delays or that business cases are always wrong. Rather, an empowered product manager acting like an *embedded CEO* (and held accountable for results) will try to fix internal problems and adjust to external constraints and needs—similar to a CEO who cannot simply excuse low performance with bad circumstances. Having worked with different companies in a variety of industries on software product management, we emphasize what we call the *4+1* best practices to optimize product management.

4+1 Product Management Best Practices

Four software product management best practices will improve the situation, if used together. These techniques have been found to reliably improve project performance. A "+1" practice is added to highlight the need for personal competence growth.

1. Install an Effective Core Team

Often, different stakeholders have unaligned agendas that make the project late and cause lots of overhead and rework. The first thing to do is formally create a core team with the product, marketing, project, and operations managers for each product (release) and make them fully accountable for the success of a product. These people represent not only the major internal stakeholders in product or solution development, but also sufficiently represent different external perspectives. The core team leads the product development in all its different dimensions. They typically meet once a week to discuss all open issues, risks, and relevant aspects of the product. Decisions are taken and implemented by the respective function. I suggest announcing and making this core team operational as early as possible in the product life cycle, but certainly when the product or release is defined. The success factor is to give this core team a clear mandate to *own* the project. I have observed that the most need for active support is in the building of an effective core team that agrees that they have to steer the course together. Too often, we face silo organizations in marketing, where product management and engineering don't work together. In many cases, this means the necessity is not only to build teams, but also to train and coach employees and to adjust annual targets

and performance management. As we often realize, culture changes when targets are adjusted.

2. Enforce the Product Life Cycle

Like the core teams, making a standardized product life cycle mandatory for all product releases (i.e., all engineering projects) is essential. Most companies today have such a life cycle defined, but rarely use it as the pivotal tool to derive and implement shared and committed decisions. Too often, requirements changes are agreed on in sales meetings without checking feasibility, and technical decisions are made without considering business case and downstream impacts. A useful product life cycle has to acknowledge that requirements may never be complete and may indeed be in a *continuum* state. The product life cycle should guide with clear criteria (i.e., determining what is good enough or stable enough). This implies that it is sufficiently flexible to handle different types of projects and constraints. This is achieved with basic tailoring techniques and guidance as to which elements are mandatory and which should be adjusted to the specific environment. To foster discipline and visibility, the mandatory elements of gate reviews (such as checklists or minutes) must be explicit and auditable. To reduce overheads, I recommend using online workflow management, which operationally embeds tools and measurements in the product life cycle. Ease of execution with such workflow automation will facilitate reuse, data quality, and consistency. With the current abundance of workflow management systems, I suggest evaluating potential solutions versus your own needs to simplify the process.

3. Evaluate Needs and Requirements

Requirements must be understood and evaluated by the entire core team to ensure that different perspectives are considered. Each single requirement must be justified to support the business case and to allow management of changes and priorities. With our clients, we often found requirements simply being *collected*, yielding lots of unnecessary features that added to complexity—but not to customer-perceived value. In fact, almost half of all delivered features are rarely used and do not provide any payback [1, 7]. If a product is developed on such an unjustified basis, it is in trouble because its requirements will continuously change. A product (release) must address a need and must have a strong business vision. This vision (i.e., what will be different with the release of the product) must be coined into a sellable story. The story then translates to business objectives and major requirements. Good product management first understands the customer's needs and business case, and then develops the necessary features. Requirements are a contract mechanism for the project internally and often for a client externally. They must be documented in a structured and disciplined way, allowing both technical as well as market and business judgment. Their evaluation should specifically look to completeness, consistency, and understandability. Ask a tester to write a test case before processing the requirement. Ask the marketing manager to check whether he or she can sell the feature as described; this avoids unrealistic or overly complex feature lists that don't address real needs. Requirements should not be overly detailed or there is a risk of *paralysis by analysis*; determine what is good enough and ensure that any further insight is adequately considered. After evaluation, requirements are approved by the core team. Only thereafter are the requirements formally allocated to the project, and the engineering effort is spent. Requirements and business objectives must be managed (planned, prioritized, agreed, monitored) throughout the life cycle to assure focus [7, 8]: Have a project plan that is directly linked with the requirements. Work packages within this project plan should show the value they contribute with such links to requirements. Following these directions allows an organization to both focus on what matters and monitor the earned value of the project from beginning to end, as well as proactively manage risks, such as effort being burned without creating value. Also note that your change management needs to be both formal and disciplined, because most issues I've seen in troubled projects result from creeping requirements and insufficient impact analysis. To ease change management, install traceability from requirements upwards to the business case and downwards to test cases.

> *"Too often, requirements changes are agreed on in sales meetings without checking feasibility, and technical decisions are made without considering business case and downstream impacts."*

4. Assure a Dependable Portfolio

Managing release roadmaps—and their own portfolio as a mix of resources, projects, and services—must be the focus of each product manager. Often, roadmaps are not worth the paper they are printed on due to continuous changes that result in a lack of buy-in from sales, operations, and service. Projects are started ad-hoc, while necessary reviews and clean-ups in portfolios rarely happen. With moving targets, sales has no guidance on how to influence clients, and engineering decides on its own which technologies to implement with what resources. The product manager has to show leadership and ensure *dependable* plans and decisions that are effectively executed. Dependable means that agreed milestones, contents, or quality targets are maintained as committed unless a change is agreed on and documented. Be aware that, as a product manager, each ad-hoc content or release change will create the perception that your portfolio is not managed well. Apply adequate risk management techniques to make your portfolio and commitments dependable; as you may find, projects may need more resources, suppliers could deliver late, or technology won't work as expected. For instance, platform components used by several products might use resource buffers, while application development applies the time-boxing technique. If there is a change to committed milestones or contents within the portfolio, it must be approved first by the core team and, where necessary, by respective steering boards and then documented and communicated with rationales.

The "+1": Evolve Your Product Management

Just having these four software management practices distilled and processes agreed upon is not sufficient in order to improve the product management

culture. Often, I've seen organizations where product managers complain about a lack of empowerment and remain in an observer role. The truth is that they simply don't have the right competencies to be empowered as a mini-business owner; this leads to the wrong people in wrong positions. To achieve a true culture change, I strongly recommend competence building for all product managers across an organization. This means change management and closely working with product managers to help them grow. Such individualized and focused competence management strengthens individual product managers and helps them achieve their missions. The equation is simple: Competence and leadership enforced from the bottom up in each project yields better products, which grows motivation and improves the overall performance.

Our software product management framework was shaped by working with hundreds of product managers worldwide in different industries [4, 5]. Figure 3 shows the product management framework in a simplified format. The top shows a product life cycle as most companies today have it formally up and running. Processes are derived from best practices and underline the formal content of product management in an organization. The middle section of the figure shows the typical processes that a product manager is responsible for, or is at least heavily involved with. Finally, what is derived from these processes (shown on the bottom of the figure) are the competency needs of an organization's product management. While there are overlaps across companies, focus areas differ (e.g., a software service provider has different focus areas in this framework than an automotive supplier).

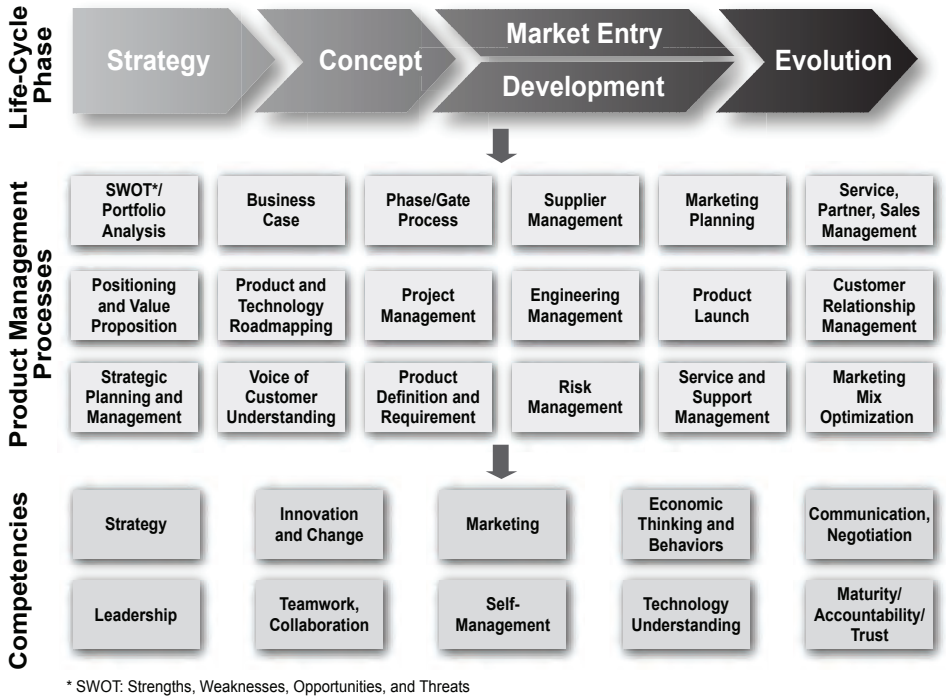
This being done, we can get back to organizational change management and working with each product manager to identify their own strengths and weaknesses. The competencies are used as a basis to provide individualized training and coaching for closing gaps. With good change management and coaching, I've observed a strong motivational push, and have seen (during the competency evolution process) the product management community starting to take shape: Incumbents had a role model (who had been actively trained); an increasing number of product managers became interested in working more methodologically, primarily because they saw the success of other business units and colleagues who had already started implementing the necessary changes [4].

Product managers often ask what they can do to deliver better results. Here are 10 ways to personally grow as a product manager:

1. Behave like an embedded CEO.
2. Drive your strategy and portfolio from market and customer value.
3. Be enthusiastic about your product.
4. Have a profound understanding of your markets, customers, and portfolio.
5. Measure your contribution on sales (top-line) and profits (bottom-line).
6. Periodically check assumptions such as business cases.
7. Take risks and manage them.
8. Foster teamwork based on Lean processes.
9. Insist on discipline and keeping commitments.
10. Be professional in communication, appearance, and behavior.

Having observed hundreds of industry projects from domains such as small software applications and services, embedded systems to large communication and IT systems, I strongly suggest applying the four software product management practices in parallel; they depend on each other. Their combined use will significantly reduce delays and thus improve market performance. These four practices are applicable in different organizations and industries. They are tangible and can be formally introduced to projects during the launch period, thus reducing the impact change and allowing an organization to see the growing benefits early in their projects.

Figure 3: *The Software Product Management Framework*



The Business Value

Does better software product management mean better business performance? At Vector Consulting Services, we have performed a root cause analysis of hundreds of products that underperformed and found similar causes reappearing. Root causes included business cases that were never re-evaluated, unbalanced portfolios that strangulate new products, insufficient management of new releases and service efforts, and a lack of vision causing requirements to continuously change. This is underlined by observations such as in [6], which indicates that the top 20 percent of enterprises deliver 79 percent of new products on-time, while the average enterprise delivers only 51 percent of on-time projects. The same holds true for efficiency: We found that with a requirements change rate beyond 20 percent in a project, productivity falls, and as such, business performance [1].

Improved product management has a profound positive impact on overall business. For instance, strengthening the product management role at Alcatel-Lucent showed that duration (time to market), schedule adherence, and handover quality all improved in a sustain-

able way. We have been working with hundreds of product managers and achieved a 20 percent per year reduction of delays [1, 4]. Explanatory factors for this positive impact of product management include leadership and teamwork, managing risks and uncertainty, mastering stakeholder needs, and accountability towards agreed business objectives—managed by one empowered person across the product life cycle.

Conclusions

Using the 4+1 method means more ownership, leadership, and motivation in product development teams and at their interfaces. Each of the practices can be applied within a single product line if a company is not yet prepared to introduce them across all product lines. The practices and overall product management framework can be gradually introduced to product lines or business units, thus reducing the change impact. Practitioners in engineering, product management, and marketing accept these practices because they yield concrete performance improvement and stimulate empowered project teams.

Growing an organization's product management discipline requires good change management to achieve a culture where these practices are used and implemented by teams across the organization, supported by their management, and communicated openly to resolve conflicts.

For improved software production and market success, product management is here to stay. It is not a proxy to arbitrate a variety of conflicting interests, but rather a key business role in an entire company that is empowered to act as a business owner. It provides the basis for success or failure in the product's development. Or, using our initial analogy: If you do not know which direction to take in cutting the trees, don't simply start just to show progress. Real progress is what creates a lasting user experience, and this is defined from a product perspective—not ad-hoc during project work in a shouting contest.◆

Acknowledgement

This article is based on evolving the product manager competence in different companies worldwide. The 4+1 best practices had been fine-tuned during many discussions at the International Workshop on Software Product Management (IWSPM) series.

References

1. Ebert, Christof, and Reiner Dumke. Software Measurement. New York: Springer, 2007.
2. Reifer, Donald J. "Profiles of Level 5 CMMI Organizations." CrossTalk Jan. 2007.
3. Gorchels, Linda. The Product Manager's Handbook: The Complete Product Management Resource. 3rd ed. New York: McGraw-Hill, 2006.
4. Ebert, Christof. "The Impacts of Software Product Management." The Journal of Systems and Software. Vol. 80, Issue 6: 850-861, June 2007.
5. IWSPM. Proc. from the International Workshops on Software Product Management. 12 Sept. 2006, Minneapolis, and 9 Sept. 2008, Barcelona, Spain.
6. Cooper, Roger G., et al. "Benchmarking Best NPD Practices." Research-Technology Management. Part I: Jan. 2004: 31; Part II: May 2004: 43; Part III: Nov. 2004: 43.
7. Davis, Alan Mark. Just Enough Requirements Management. New York: Dorset House, 2005.
8. Karlsson, Lena, et al. Challenges in Market-Driven Requirements Engineering – An Industrial Interview Study. Proc. of 8th International Workshop on Requirements Engineering: Foundations for Software Quality. Essen, Germany: 37-49. 9-10 Sept. 2002 <www.tts.lth.se/Personal/bjornr/Papers/REFSQ02.pdf>.

About the Author

Christof Ebert, Ph.D., is managing director and partner at Vector Consulting Services. He is helping clients worldwide to improve technical product development and to manage organizational changes. Prior to working at Vector, he held engineering and management positions for more than a decade in telecommunication, IT, and transportation. As a business consultant, author of several books, lecturer at the University of Stuttgart, and public speaker, he has influenced numerous companies with his results-driven contributions.

Vector Consulting Services
Ingersheimer Straße 24
D-70499 Stuttgart
Germany
Phone: +49-711-80670-0
E-mail: christof.ebert@vector-consulting.de