



SPM Syllabus Foundation Level

V.1.3

This syllabus was written by the following members of the International Software Product Management Association (ISPMA): Hans-Bernd Kittlaus (editor), Jonas Als, Magnus Billgren, Erik Bjernulf, Sjaak Brinkkemper, Christof Ebert, Samuel Fricker, Tony Gorschek, Rainer Grau, Gerald Heller, Marc Hilber, Barbara Hoisl, Slinger Jansen, Mahvish Khurum, Daniel Lucas-Hirtz, Andrey Maglyas, Kevin Vlaanderen and Inge van de Weerd.

We thank all honorary authors and contributors.

Copyright © 2016 for this syllabus is with the authors mentioned above. The rights have been transferred to the ISPMA International Software Product Management Association. Licensing by other standardization bodies is possible upon written acceptance.

This document is provided for educational purposes. ISPMA does not warrant that it is suitable for any other purpose and makes no expressed or implied warranty of any kind and assumes no responsibility for errors or omissions. No liability is assumed for incidental or consequential damages in connection with or arising out of the use of the information contained herein.

Terms of Use:

1. Individuals and training providers may use this syllabus as a basis for seminars, provided that the copyright of the authors and ISPMA is acknowledged and included in the seminar materials.
2. Any individual or group of individuals may use this syllabus as a basis for articles, books, or other derived publications, provided that the copyright of the authors and ISPMA as the source and owners of this document is acknowledged in such publications.

Any inquiries regarding this publication, requests for usage rights for the material included herein, or corrections should be sent by email to info@ispma.org.

Preface

The goal of the International Software Product Management Association (ISPMA) syllabus for the ISPMA Certified Software Product Manager is to promote general understanding of the *discipline of product management for software products* including the management of software parts of software-intensive products, i.e. systems or services.

The syllabus “SPM Foundation Level” covers the full spectrum of elements of software product management that are well-supported by literature and industrial practice. The syllabus corresponds to a 3-day industry course.

The syllabus addresses the needs of people involved in software product management, including those that the product manager interfaces with, e.g. general management, marketing and sales, research and development, production, service and support, and controlling.

The syllabus is the basis for examination to certify that the examinee has achieved the degree of knowledge described in this syllabus. The terms used in this syllabus are consistent with the glossary of the ISPMA.

Purpose and structure of the Syllabus:

The syllabus is the basis for consistent training, learning, and examination of software product management. It provides:

- Explicitly phrased educational objectives for each chapter, and
- Informal explanations to detail the educational objectives.
- Informal references to literature (without limiting the interpretation of the syllabus to this literature only).

This syllabus consists of five chapters. Each chapter covers one major educational unit (EU). Each chapter also includes the duration suggested to teach it. Each educational unit has educational objectives (EO) that are enumerated following the chapter headers (EO1.1.1, EO1.2.1 ...). An educational objective has a defined cognitive level of knowledge that the course participant is expected to achieve. The numbering scheme for these objectives is aligned with the chapter numbering.

The educational objectives are expressed in terms of two cognitive levels of knowledge phrased using the verbs “knowing” for level 1 and “understanding” for level 2. These two verbs are placeholders for the following:

- L1 (know): enumerate, characterize, recognize, and name.
- L2 (understand): reflect, analyze, execute, justify, describe, judge, display, design, develop, complete, explain, elucidate, elicit, formulate, identify, interpret, reason, translate, distinguish, compare, understand, suggest, and summarize

Each EO in the syllabus has one or both of the two cognitive levels assigned to it.

Included and excluded key areas:

The syllabus covers knowledge applicable for any kind of software systems and organizational contexts. A training course may cover more domain-specific details if needed by the course participants. This syllabus, however, does not provide guidance for such specialization, rather describes the base knowledge necessary, which can be complemented with domain specific items. An addendum to the syllabus discusses different business scenarios (i.e. software intensive systems

and software services) and outlines specifics with regard to the application of the syllabus contents in the different scenarios.

The syllabus is independent of any specific process model, and thus defines knowledge of a software product manager without prescribing exact interfaces to other roles in a product organization.

Training Courses:

The syllabus corresponds to a three-day industry course. The syllabus does not prescribe the specific form and approach of learning, however. It can also be administered with other forms of teaching or learning, such as self-learning supplemented by coaching or courses at universities or universities of applied sciences.

Training providers are encouraged to tailor training courses to the participants, and to add examples and appropriate exercises that give participants an opportunity to apply the training contents to practical cases. A participant should carefully choose the training provider. A list of training providers can be found on the ISPMA web site www.ispma.org.

Examination:

The syllabus is the basis for the examination for the ISPMA foundation-level certificate “ISPMA Certified Software Product Manager”. All chapters are relevant for the exam with the exception of the addendum. The exam takes the following form:

- Demonstration of knowledge with a multiple-choice test

Multiple-choice tests can be held immediately after a training course, but also independently from courses (e.g. publicly announced exams of the examination authorities). A list of accredited examination authorities can be found on the ISPMA web site www.ispma.org.

Course participant prerequisites:

The training and learning of the syllabus assumes general knowledge of, and some experience in, the management or development of software products or software in software-intensive systems.

The formal background of the course participant is not crucial (whether it be engineering or management), rather the level of experience is predominantly the factor associated with the prerequisites. A course participant should have at least three years of experience in the software industry in general. However, this is a generic recommendation and might not be applicable for all situations or course participants.

Table of Contents

EU1	Introduction and Foundations	
	1.1 Software Product Management Essentials	1:30 h
	- Software as a Business	
	- Relevant terms, role, objectives	
	1.2 Software Product Management Framework	1:00 h
	- Software Product Management Overview	
	- Structure of the Framework	
EU2	Product Strategy	
	2.1 Product Strategy Essentials	2:00 h
	- Role and Elements of Product Strategy and Their Interdependencies	
	- Product Name	
	- Positioning and Product Definition	
	- Delivery Model, Service Strategy and Sourcing	
	2.2 Business Aspects	2:30 h
	- Business Models	
	- Business Case	
	- Costing and Pricing	
	- Performance and Risk Management	
	2.3 Ecosystem Management	0:45 h
	- Role and Elements	
	- Stakeholders	
	2.4 Legal Aspects	0:45 h
	- Contracts	
	- Management of Intellectual Property Rights (IPR)	
EU3	Product Planning	
	3.1 Product Requirements Engineering	1:30 h
	- Role of Requirements Engineering in Software Product Management	
	- Inquiry cycle with elicitation, analysis, and validation	
	3.2 Release Planning	1:00 h
	- Release Planning Process and its conflicts	
	- Structure of Release Plan	
	3.3 Roadmapping	1:00 h
	- Product Roadmap and its elements	
	- Sources of input	
	- Usage of Roadmaps	
	3.4 Product Life Cycle Management	1:00 h
	- Phases of the Life Cycle	
	- Performance Management	
	3.5 Impact From Development Methodologies	0:30 h
	- Agile, Iterative, and Waterfall Development Methodologies	

EU4	Strategic Management	2:00 h
	<ul style="list-style-type: none"> - Strategic Management Essentials - Corporate Strategy and Portfolio Management - Innovation and Resource Management - Market and Product Analysis 	
EU5	Orchestration of the Organization's Functional Areas	
	5.1 Development	0:45 h
	<ul style="list-style-type: none"> - Engineering Management - Project Management incl. Agile Methodologies - Project Requirements Engineering - User Experience Design - Quality Management 	
	5.2 Marketing	0:30 h
	<ul style="list-style-type: none"> - Marketing Planning - Customer Analysis and Opportunity Management - Marketing Mix Optimization - Product Launches and Operational Marketing 	
	5.3 Sales and Distribution	0:30 h
	<ul style="list-style-type: none"> - Sales Planning and Channel Preparation - Customer Relationship Management - Operational Sales and Distribution 	
	5.4 Service and Support	0:30 h
	<ul style="list-style-type: none"> - Service Planning and Preparation - Service Provisioning - Technical Support - Documentation - Marketing and Sales Support 	
	5.5 The Role of SPM as Orchestrator	1:00 h
	<ul style="list-style-type: none"> - Roles and Responsibilities of SPM - Coordination Challenges and Strategies - Additional Areas (SPM, Finance, Controlling, HR) 	

A. Addendum: SPM in Different Business Scenarios

Bibliography

EU1 Introduction and Foundations

Duration: 2:30h

Educational Objectives:

- EO1.1.1 Know the scope and effects of software product management.
- EO1.1.2 Know the role and objectives of software product management and relevant terms.
- EO1.2.1 Know the SPM Framework.
- EO1.2.2 Understand the multi-disciplinary nature of SPM.

1.1 Software Product Management Essentials

Duration: 1:30 h

Product management has become an established discipline in many industries since Procter & Gamble introduced it in 1931. During the last decades, many software product companies - such as Microsoft and IBM - implemented Software Product Management (SPM) in their organizations, as did corporate information technology organizations in other industries, and companies that produce software embedded in software-intensive products and services. The role of software product manager has emerged during this time as being of strategic value, since it is crucial for the economic success of a product. Successful product management means delivering the right products at the right time and price for the right markets, and increasing profitability, product quality, customer satisfaction, and the success rate of release projects in terms of schedule predictability and time-to-market.

Literature: *C. Ebert (2007); S. Fricker (2012); H.-B. Kittlaus, P. Clough (2009, p. 5 ff)*

A product is a combination of goods and services, which a supplier/development organization combines in support of its commercial interests to transfer defined rights to a customer. A software product is one whose primary component is software. It compares to non-software products in terms of high complexity, negligible manufacturing cost, great flexibility, and a high rate of change. In this syllabus, software product management means the management of software products and software parts of software-intensive products, i.e. systems or services. Software parts of software-intensive systems that are not marketed and priced as separate entities are called embedded software. Software-intensive systems can be products from all industries like cars, airplanes, smartphones etc. Software-intensive services, often delivered as cloud or internet services, can also be products from all industries like financial, insurance, gaming, social software, or personal services based on software support. To support ease of reading this syllabus will generally use the term "software products" instead of explicitly referring to software-intensive products as well.

Literature: *H.-B. Kittlaus, P. Clough (2009, p. 5 ff)*

A product family is a set of products with common features, integration, interface similarities, and technological similarity to address a specific problem. It is defined for marketing reasons. A product line is based on a platform with defined (static or dynamic) variability for developing products tailored to different markets and users, while increasing quality and decreasing cost. Its definition is based on both technical and business reasons.

Literature: *H.-B. Kittlaus, P. Clough (2009, p. 10 f)*

A software product manager is responsible for managing software with the objective to achieve sustainable success over the life cycle of the software product (family or line). This generally refers to economic success, which is ultimately reflected by the profits generated. Software product managers have the business responsibility across different versions, variants and associated services of a product. They act as a “mini CEO”, i.e. they have to manage a broad set of product-related activities as shown in the ISPMA SPM Framework (Figure 1). Typically they have direct responsibility for the activities marked as “Core SPM”, in particular “Product Strategy” and “Product Planning”. For the activities under “Strategic Management”, software product managers participate by representing their products on the corporate level, e.g. in portfolio management, by providing input and making use of the results. For the activities under “Development”, “Marketing”, “Sales and Distribution” and “Service and Support”, the direct responsibility is typically with other units in the company, but software product managers have to orchestrate these activities such that they are performed in line with product strategy and plan. Given the broad set of responsibilities, prioritization is needed on an ongoing basis and can be based on the respective estimated impact on short and long term profitability.

Literature: C. Ebert (2007); S. Fricker (2012); H.-B. Kittlaus, P. Clough (2009, p. 39 ff)

1.2 Software Product Management Framework

Duration: 1:00 h

The SPM framework provides a holistic view on the **elements** of software product management. It can be used as a model to establish and improve the discipline of software product management in an organization. It is structured in the following way:

- The horizontal structure (columns) is based on the functional areas of a software organization.
- Vertically, i.e. within the columns, the structure is based on a top-down approach, i.e. from strategic and long-term to operational and short-term. However, the interdependencies of the elements within each column (and also across columns) are more complex than can be fully expressed in a two-dimensional structure. There are a number of cases where the actual doing requires iterative processes that go back and forth between elements until everything is synchronized. A good example is the Product Strategy column where this kind of iterative approach is mandatory between most elements before a product manager gets to a consistent strategy. Also, there are elements like "Ecosystem Management" and "CRM" that contain both longer-term and shorter-term aspects.
- There is an additional overlay structure with "Core SPM", "Participation" and "Orchestration". For Market Analysis and Product Analysis in the Strategic Management column, corporate functions are typically responsible in larger companies with the product manager participating, in smaller companies the product manager may be responsible. In any case, getting reliable information on market and product on a frequent basis is part of the core SPM responsibilities.

Strategic Management	Product Strategy	Product Planning	Development	Marketing	Sales and Distribution	Service and Support
Corporate Strategy	Positioning and Product Definition	Product Life-Cycle Management	Engineering Management	Marketing Planning	Sales Planning	Service Planning and Preparation
Portfolio Management	Delivery model and Service Strategy	Roadmapping	Project Management	Customer Analysis	Channel Preparation	Service Provisioning
Innovation Management	Sourcing	Release Planning	Project Requirements Engineering	Opportunity Management	Customer Relationship Management	Technical Support
Resource Management	Business Case and Costing	Product Requirements Engineering	User Experience Design	Marketing Mix Optimization	Operational Sales	Marketing Support
Market Analysis	Pricing		Quality Management	Product Launches	Operational Distribution	Sales Support
Product Analysis	Ecosystem Management			Operational Marketing		
	Legal and IPR Management					
	Performance and Risk Management					
Participation	Core SPM		Orchestration			

Fig. 1 SPM Framework V. 1.2

This SPM Foundation Level Syllabus is structured in correspondence with the SPM Framework with only a few deviations due to pedagogical reasons. It starts with the core SPM columns Product Strategy and Product Planning, then Strategic Management and the Orchestration of the Functional Areas are described.

Literature: C. Ebert (2007); S. Fricker (2012); H.-B. Kittlaus, P. Clough (2009, p. 43 ff); I. van de Weerd, S. Brinkkemper, R. Nieuwenhuis, J. Versendaal, L. Bijlsma (2006)

EU2 Product Strategy

Duration: 6:00 h

Educational Objectives:

- EO2.1.1 Understand the role of a product strategy, its elements and their interdependences.
- EO2.1.2 Know the relationship of product strategy to corporate strategy and portfolio management.
- EO2.1.3 Understand the software product manager's role as "mini CEO".
- EO2.2.1 Understand the business aspects and their interdependences.
- EO2.2.2 Understand the business model concept and the business model canvas.
- EO2.2.3 Understand business cases.
- EO2.2.4 Know the Strategic Pricing Pyramid.
- EO2.3.1 Know the role, significance and elements of a software ecosystem.
- EO2.3.2 Know the relevant stakeholders of a software ecosystem.
- EO2.4.1 Understand the legal differences between a software license contract and a service contract.
- EO2.4.2 Understand the importance of protecting intellectual property.
- EO2.4.3 Know intellectual property protection mechanisms.

2.1 Product Strategy Essentials

Duration: 2:00 h

Software product managers are responsible for defining the strategy for their product (or platform or family) and for supporting and updating it over time. Normally, a strategy covers a time span of about one to five years, however this varies in relation to domain. The product strategy describes how the product is supposed to evolve over this strategic timeframe. It should include the following items:

- Product vision
- Product definition
 - functional scope
 - quality scope
 - intended use and users
 - user experience (UX) design scope
 - complete set of components that determine the offering

- Target market, potential segments
 - Scope, size, competitive products and their market shares
 - International markets and their opportunities
 - Segmentation
- Delivery model and Service Strategy
 - Licensed product vs. service offering (e.g. Software-as-a-Service (SaaS))
 - Standard product vs. customizable product
 - Packaging (shrink-wrapped, download, combination with services etc.)
 - Services that are part of the total offering
- Product positioning
 - Value definition (from a customer perspective)
 - Focus with regard to target market and segments, company product portfolio, and product life cycle phase (e.g. revitalization)
 - Channel options
 - Required partnerships and alliances
- Sourcing
 - Options and their pros and cons
 - Make or buy decisions
- Business plan
- Roadmap

Very often, this content is described in one cohesive product strategy document in order to emphasize the need for full consistency. The product strategy column in the ISPMA SPM Framework (see Fig. 1) contains additional elements, i.e. Pricing, Ecosystem Management, Legal and IPR Management, and Performance and Risk Management, which are also of strategic importance, but typically not fully included in a product strategy document.

All these items are highly interdependent. If, for example, business planning results in an available budget smaller than originally assumed, it will only be possible to expand the product scope to a lesser extent or more slowly. If new segments are to be added to the target market within the strategic time frame, the product scope may have to be expanded. Dependency on other products can also have considerable consequences, e.g., if certain functionalities or enabling code must be available in several products at the same time. In bigger companies that have one or several product portfolios an individual product strategy needs to be aligned with the corporate strategy and portfolio. It should be observed that interdependencies can exist on different levels of abstraction, ranging from portfolio to product to feature to function to component and also covering management and business decisions included in the strategic concerns described above.

To convince and engage all stakeholders, a strong product vision is helpful. Basically, it is a conceptual image of what the future product will be, i.e. high-level descriptions of a product concept

(What is it? How can the product satisfy the needs of potential customers better than competition?) and a business model (How can the product be built, sold and supported in a profitable way?). The other elements of the product strategy provide the details that turn the vision into a manageable and executable path into the future. In bigger companies a product vision needs to be aligned with the corporate vision.

Literature: *H.-B. Kittlaus, P. Clough (2009, p. 53 ff); M. McGrath (2000)*

2.2 Business Aspects

Duration: 2:30 h

The primary objective of product management is to achieve sustainable success over the life cycle of the product (or family or line). This generally refers to economic success, which is ultimately reflected by the profits generated. Since profits lag behind investments, i.e., an investment phase involving losses will be followed by an extended profitable phase, a longer-term perspective is appropriate. Therefore the product manager has the role of a mini CEO who has to plan and keep track of the business aspects. This includes the representation of the product on the corporate level.

A business model describes the rationale of how an organization creates, delivers and captures value. It is often considered at the corporate or business unit level, but its consideration can also make sense on a product level, in particular when a product requires a business model that has not yet existed in the company before. The Business Model Canvas is one of the tools that have been used to review and challenge existing business models and systematically invent new ones that change the way a product competes. It defines and visualizes a business model by integrating positioning and product definition, delivery model and service strategy, pricing, sourcing, business case and costing, and ecosystem which are the product strategy cells of the ISPMA SPM Framework.

Literature: *A. Osterwalder, Y. Pigneur (2010); S. Chen, A. Cheng, K. Mehta (2013)*

The business aspects refer to the following subjects:

- Definition and selection of relevant business measures
 - Possible measures: Profit, revenue, market share, no. of active customers, no. of installed licenses, conversion rates, cost/budget, customer satisfaction, customer perceived value, competitor gap, innovativeness etc.
- Performance management
 - Continuous tracking and analysis of selected relevant measures; timely action taking if needed
- Risk management
 - continuous tracking and analysis of risks identified in connection with development, sales and customer use of the software product; timely action taking if needed

- Business planning
 - Forecasting the selected relevant measures over the strategic timeframe
 - Planning of resources and budgets
- Costing
 - Estimation methods
 - Target costing
- Business case
 - Approaches for the comparison of quantified estimates for cost and benefit
 - Scenarios in which business cases are relevant

Literature: *H.-B. Kittlaus, P. Clough (2009, p. 62 f); M. Schmidt (2002)*

- Pricing
 - The importance of price with regard to business success, i.e. in relation to the relevant measures
 - Market- and value-based pricing
 - Problems of cost-based pricing for software offerings
 - Strategic Pricing Pyramid (price strategy, policy, level)(see figure 2)
 - Typical pricing models for software including freemium

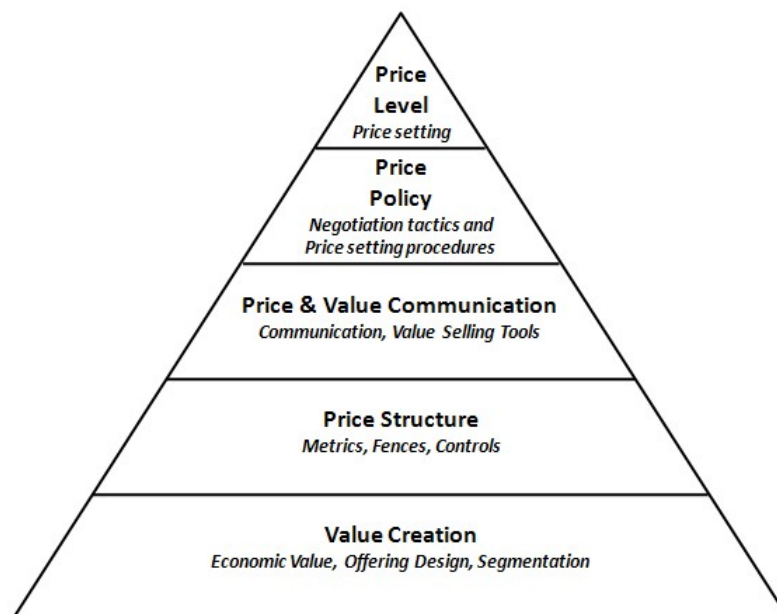


Fig. 2 Strategic Pricing Pyramid (© Nagle/Hogan 2005)

Literature: *H.-B. Kittlaus, P. Clough (2009, p. 117 ff); T.T. Nagle, J.E. Hogan, J. Zale (2014); T.T. Nagle, J.E. Hogan (2005); M.N. Sodhi, N.N. Sodhi (2007)*

A business case is a decision-support and planning approach that compares likely financial results and other business consequences with the required investment for a given undertaking, in the case

of software typically a development and marketing effort. To be effective, the business case should communicate the following information: the description of the undertaking including the underlying assumptions, an estimate for the required investment, the approach to generate business benefits with impact on earnings over time, and a sensitivity, risk, and contingency analysis. A business case is credible when it is complete, balanced with important scenarios elaborated, based on best practice methodology, and with underlying assumptions made explicit and accepted by all stakeholders.

Literature: *M. Schmidt (2002)*

2.3 Ecosystem Management

Duration: 0:45 h

A software ecosystem is defined as a set of businesses functioning as a unit and interacting with a shared market for software and services, together with relationships among them. These relationships are frequently underpinned by a common technological platform and operate through the exchange of information, resources, and artifacts. Organizations can take on different roles in ecosystems, such as keystone, dominator, or niche player.

Product managers are greatly influenced by the role its organization wishes to play in a software ecosystem, and the extended network of the organization is of increasing importance for the daily work of the product manager. How, for instance, must the product manager of a platform deal with a partner that delivers a feature that directly competes with the platform company's features? And how must platform owners deal with incorrect use of its APIs or extensive feature requests from partners?

It is usually not part of the responsibilities of a software product manager to decide on the strategy and role that his/her company wants to play in an ecosystem, as these decisions are typically left to dedicated channel managers or corporate strategy makers, i.e., executive management. However, once these decisions are finalized, they have a significant impact on the work of the product manager. The product manager must conform to the role his/her company wishes to play in the ecosystem which has direct influence on the positioning of the product, the pricing strategy, the degree of dependence of road map and requirement decisions on other players in the ecosystem, etc. In addition to this, a product manager can give input as to how the ecosystem strategy works, and suggest opportunities for the evolution of stated strategy.

The structural constituents of software ecosystems are stakeholders, relationships, boundaries, behavior, and strategies. Software ecosystems typically emerge around major products or technology trends. Stakeholders can be:

- Software vendors (components, platforms, plug-ins, add-ons)
- Marketing and Sales partners (Value Added Resellers (VARs), Independent Software Vendors (ISVs), Original Equipment Manufacturers (OEMs), System Integrators (SIs), consultants)
- Influencers (market research companies, journalists, consultants)

The collaboration can regard requirements, development, marketing, sales, distribution, and services. Partnerships and alliances that are required to actually provide a complete offering to the customer need special attention.

Literature:

J. Bosch (2009); M. Iansiti, R. Levien (2004); S. Jansen, M. Cusumano, S. Brinkkemper (2013); H.-B. Kittlaus, P. Clough (2009, p. 25 ff, p. 63f); D.G. Messerschmitt, C. Szyperski (2003); G.A Moore (2014); K.M. Popp, R. Meyer (2010)

2.4 Legal Aspects

Duration: 0:45 h

Software product managers need to consider several legal aspects specifically related to software products. Details are typically handled by legal experts (e.g. counsels), but product managers need to have an overview of the legal risks they bear in their role as “mini CEO”. First there are contractual issues between the software vendor and the customer. Then there is the protection of intellectual property. Thirdly there are specific risks such as governance, finance, supply-chain, delivery commitments, product liability, data protection (especially for SaaS), laws on general terms and conditions, blacklisting of countries for specific software components etc. which are increasingly relevant (see for instance the growing impact of governance rules and transparency laws).

The contract by which software is “acquired”, maybe individually negotiated or, particularly in the mass-market, based on so-called “terms and conditions,” which describe the generally applicable legal terms, including:

- Scope of the license or service
- Guarantee and Warranty / SLA
- Transferability
- Type of charges
- Liability
- Maintenance provisions (a separate maintenance agreement may be concluded.)
- Miscellaneous legal provisions (e.g. set-off, default, dispute resolution, governing law, severability clause)

Terms and conditions are accompanied by further documents which cover product-specific terms, such as a transaction document which may contain a definition of the customer enterprise, number of licenses acquired, prices, terms of payment, contract duration (if any) and termination (if any).

The term “license” is a non-technical legal term which is not described by statute. Therefore, the understanding of what a license is may differ, in particular from jurisdiction to jurisdiction. On a common sense basis, a license describes to which extent and under which conditions the licensee can use an item, e.g. software, which is subject to intellectual property rights, in particular trademarks, patents or copyrights. These rights of use are granted by the licensor which may be the owner of all rights in such item or may be an entity authorized to grant the license, e.g. a reseller.

The term “license contract” means an agreement between the licensor and the licensee about all terms in connection with one or more licenses. With software, the term “license” usually describes the scope of rights of use which the licensor grants to the licensee. Such rights may be exclusive, or they may be limited in some way, e.g. geographically or timely. Software licenses may also allow only a particular way of using the software, e.g. only on premise or also for Software-as-a-Service offerings. In the case of SaaS or other software-intensive services, the service provider needs license contracts with licensors of the software that is used as part of the service offering which explicitly allow this kind of use. Customers of such a service do not need license contracts, but only service contracts with the service provider.

Literature: *H.-B. Kittlaus, P. Clough (2009, p. 64 ff)*

Since the development of software requires significant investment, and software can be easily copied, it is of utmost importance to the investor that the intellectual property resulting from the investment is protected. There are four fundamental legal constructs for the protection of intellectual property:

- Trademark: Protection for the names of brands, i.e. brands do not apply to the software itself but only to the brand under which it is marketed.
- Trade Secret: Protection of company-internal knowledge (primarily against employees). This protection is exercised by restricting knowledge and access to a very small number of people and by using non-disclosure agreements. In most jurisdictions, trade secrets are only protected under unfair competition laws.
- Copyright: Protection against copying of software code (as specific expressions of an idea or way of doing something) and product material such as manuals, brochures and presentations. This is the main way software is protected. The algorithm or idea behind software is not protected under copyright law.
- Patent: Protection of the specific technical concept or idea. In most jurisdictions, patent protection can only be obtained for software which is integrated into a technical solution to a problem.

Literature: *H.-B. Kittlaus, P. Clough (2009, p. 73 ff); B. Klemens (2006);*

EU3 Product Planning

Duration: 5:00 h

Educational Objectives:

- EO3.1.1 Understand the role of requirements engineering in software product management.
- EO3.1.2 Understand the differences between product, project, and customer RE.
- EO3.1.3 Know the inquiry cycle with elicitation, triage, analysis, selection and validation.
- EO3.1.4 Know how to specify requirements.
- EO3.1.5 Know the generic life cycle model of requirements.
- EO3.2.1 Know the purpose of release planning.
- EO3.2.2 Understand the conflicting interests in the release planning process.
- EO3.3.1 Know the basic elements of a roadmap.
- EO3.3.2 Know the major sources for inputs into a roadmap.
- EO3.3.3 Understand the various ways roadmaps are used and communicated.
- EO3.4.1 Know the phases of the product life cycle.
- EO3.4.2 Understand the changes in product management's focus from phase to phase.
- EO3.5.1 Understand the impact of a chosen development methodology on the interface between software product management and development.

3.1 Product Requirements Engineering

Duration: 1:30 h

Requirements engineering (RE) in a software product management context covers the classical requirements engineering activities, such as elicitation, specification, validation, and maintenance, adapted to the market-driven situation with many often anonymous customers, competitors, and suppliers. Three requirement types can be distinguished:

- Customer requirements which express individual customer needs. They are typically under Sales' responsibility as part of their Customer Relationship Management responsibility.
- Product requirements which address no individual customer, but rather one or more markets consisting of any number of customers that may have similar needs, but with high variability. This market focus influences and impacts all other aspects of product RE, especially elicitation of requirements, innovation candidates, decisions, and management. Product RE is part of the SPM's responsibilities, and tightly linked to specific SPM activities needed in a market-driven context, such as product visioning, strategy definition, roadmapping, innovation, and release planning. Product requirements often result in requirements for development projects, but there are also product requirements that address other areas, e.g. bundling, new pricing models, new delivery processes, or improved support.

- Project requirements which further refine product requirements that are selected for a particular release or project and require implementation in software code. Project requirements can also address development-internal needs that are raised during the course of a project. Project RE is under Development's responsibility.

Literature: *H.-B. Kittlaus, P. Clough (2009, p. 79 ff)*

A product organization needs a dedicated process to deal with requirements in a systematic way, the „requirements inquiry cycle“, which combines requirements elicitation, triage, analysis, selection and validation.”

Requirements elicitation is the systematic application of methods such as interviews, focus groups, workshops, observation, creativity, surveys, and artifact analysis to understand the application domain and to identify stakeholders with their objectives and expectations. Important sources of requirements are internal; in particular research and development can ensure innovation. The resulting requirements need to be properly documented, either in natural language or model-based.

Triage means a first analysis and decision if a given requirement is inside or outside of the product scope, i.e. needs more detailed analysis or not. This may include a bundling of small low-level requirements in order to increase the productivity of the process. The product manager will typically maintain a backlog with a significant number of requirements that are inside the product scope, but not yet implemented, and that need frequent reevaluation.

Requirements analysis aims at matching the elicited information with appropriate solution concepts by applying specification, modeling, and prototyping methods which requires decision making. It includes estimation of cost and required resources.

Selection means the decision making which requirements are implemented in a particular product release, i.e. selection includes prioritization and is part of release planning.

Requirements validation involves the application of methods such as reviews, inspections, and simulations to ensure that the proposed solution adequately takes the problem context into account and is acceptable to its stakeholders. The result of successful requirements engineering is a specification that documents agreement between stakeholders and development or suppliers of what will be delivered.

Literature: *T. Gorschek, S. Fricker, K. Palm, S. Kunsman (2010); M. Khurum, A. Khurum, T. Gorschek (2007); K. Pohl, K. Rupp (2011); B. Regnell, S. Brinkkemper (2005); D. Zowghi, C. Coulin (2005)*

Several requirement life cycle phase models exist. Examples of phases that might be used are:

- *New*: the initial state of a requirement.
- *Approved*: the requirement is approved and ready to be analyzed.
- *Specified*: the requirement has been analyzed with regard to implementation concept and cost and impact estimates.
- *Rejected*: the requirement is a duplicate, already implemented, or out of scope.
- *Selected*: the requirement has been selected for implementation with a given priority.

- *Implemented*: the requirement has been implemented, i.e. development is finished.
- *Tested*: the necessary tests have been carried out in order to ensure an adequate level of quality.
- *Released*: all activities for the product release have been completed.

However, the naming and number of phases may vary, depending on the preferences of the organization.

Literature: *K. Pohl, K. Rupp (2011); B. Regnell, S. Brinkkemper (2005)*

In order to be able to manage and communicate the requirements process, traceability is required, i.e. the ability to follow the path from the original requirement to the software implementation and back.

Literature: H.-B. Kittlaus, P. Clough (2009, p. 79 ff)

Needs and expectations arrive in different shapes and forms to development organizations and are called different names, depending on their abstraction level. Goals document desired impact of the solution on stakeholders and the organization and relate the software to organizational and product strategies. A scenario describes a concrete example of satisfying or failing to satisfy a goal (or a set of goals). Requirements are capabilities of a solution that are needed by a stakeholder to solve a problem or achieve an objective. Functional and quality (sometimes referred to as non-functional) requirements are often differentiated. A third type of requirement are constraints, which are business, project or design decisions taken in advance to ensure the solution fits business, managerial and contextual concerns.

Literature: *T. Gorschek, C. Wohlin (2006); K. Pohl, K. Rupp (2011); B. Regnell, S. Brinkkemper (2005)*

Requirements often are specified in standardized documents or in the form of sorted lists. For each requirement, standard attributes are used. To address misunderstandings, language templates such as user stories can be used to specify functional requirements in natural language. To further increase the precision of a specification, semi-formal graphical specification can be employed, e.g. use case diagrams to summarize the services of a system provided to its direct context. Non-functional requirements can be specified qualitatively, by example, operationally, or quantitatively.

Literature: *K. Pohl, K. Rupp (2011); B. Regnell, S. Brinkkemper (2005)*

3.2 Release Planning

Duration: 1:00 h

Due to marketing and management reasons, a software product evolves and exists in different releases. The long-term sequence and rough contents of releases and versions is planned and documented in the roadmap. Dependent on the type and amount of change, releases are called major, minor, update, or service (patch) release. Major releases are often - for marketing reasons - called versions. Release planning is concerned with defining the detailed contents of one forthcoming product release in an attempt to maximize the value of the release in relation to the product's success over its life cycle. It is tightly linked to product RE.

The release planning decisions balance opposed forces. On one side, the selected requirements need to satisfy business objectives and real customer needs, while leading to a recognizable advantage over competition. There needs to be a balance between technology push and market pull, i.e. between innovative features and customer requirements. On the other side, the selected requirements need to account for the capabilities and capacity of the product organization, while being compliant with time and budget constraints and architectural considerations. Other influencing factors can be customer commitments and sales and marketing activities like fairs. Release planning involves negotiations and setting priorities to resolve conflicts between stakeholders about release contents and interests that are pursued with the evolving product. The release planning decisions ought to be based on strategic guidelines, e.g. to what degree the company is reactive to its markets by giving priority to customer needs or pursues a proactive innovation strategy by pushing new technologies to the markets. Results are documented in a release plan that all stakeholders finally have to agree to. Typically this is an iterative decision-making process involving the stakeholders in requirements engineering and release planning. The software development approach can also have an impact since agile development requires the software product manager's ongoing involvement in decisions about release contents and synchronization with release planning.

Literature: *P. Berander, A. Andrews (2005); P. Carlshamre, K. Sandahl, M. Lindvall, B. Regnell, J. Natt och Dag (2001); M. Cohn (2006); S. Fricker, S. Schumacher (2012); L. Lehtola, M. Kauppinen, J. Vähäniitty (2007); G. Ruhe (2010); C. Wohlin, A. Aurum (2005)*

3.3 Roadmapping

Duration: 1:00 h

Product roadmapping translates the product strategy into a series of releases that satisfy the business goals of the company and cover the strategic timeframe, i.e. between one and five years. A product roadmap usually has the following basic elements:

- Timescale
- Releases and versions
- Release themes and main features
- Target markets
- Product dependencies
- Technology impacts

The timing of releases is often governed by the “release heartbeat”, i.e. a standard frequency the software product manager defines for major and minor releases. Inputs for the roadmap come from release planning, requirements engineering, roadmaps of other products either from the company itself or from partner companies, and from product strategy. There may be separate technology roadmaps that a product roadmap needs to be synchronized with.

Product roadmaps are constructed for internal and external audiences. Internal roadmaps set the scope for specific product releases. They provide the basis for forecasting, budgeting and the instantiation of projects for the development of specific product releases. They also help with the alignment of product strategies within a company’s portfolio. External roadmaps are used to establish a communication channel between customers and the company. They are a means to communicate strategy, receive feedback on it, and build trust in the commitment of the company to long-term continuous investment in the product. High level release themes from the roadmap are key elements to guide the release planning process.

The relationship between roadmapping and release planning is neither top-down nor bottom-up, but the two are combined through an iterative process.

Literature: *R. Kostoff, R. Schaller (2001); R. Phaal, C. Farrukh, D. Probert (2007); R. Phaal, C. Farrukh, D. Probert (2004)*

3.4 Product Life Cycle Management

Duration: 1:00 h

Product management is responsible for a product along the entire life cycle. Each phase of the life cycle has its individual characteristics, focus areas and leading stakeholders besides product management (Table 1). In the first three phases, investments are necessary to develop the product. During the maturity and decline phases, the product serves as a cash cow, i.e. it generates significant revenue with rather little investment. The resulting profit can be used to finance other promising products in the portfolio.

Phase	Focus Areas	Leading Stakeholders
Conception and creation	Innovation, positioning, investment	Research, Development, Marketing, regulatory bodies
Market Introduction	Launch, market share, investment	Marketing, Sales
Growth	Market share, functionality, investment	Research, Development, Marketing, Sales
Maturity	Revitalization, service	Sales, Services, Support
Decline	Customer retention	Marketing, Sales, Support
Withdrawal	Customer retention, cost reduction	Marketing

Table 1: Software Product Life Cycle

Product management must have a solid understanding of the various phases in order to develop strategies and activities that optimally support a product in a specific phase. This requires tight cooperation with the involved functional units within the company.

The life cycle responsibility encompasses product-related knowledge management, i.e. the product manager needs to ensure that the knowledge required for the viability of the product continues to be accessible and available for the company during the product life cycle.

Continuous measurement of a product's performance is a prerequisite to drive corrective actions if needed. A product manager therefore must monitor and analyze how well the product is performing including product profitability, actual versus planned revenue, customer satisfaction, and market share.

Literature: *C. Anderson, C. Zeithaml (1984); V. Rajlich, K. Bennett (2000)*

3.5 Impact From Development Methodologies

Duration: 0:30 h

Development organizations use a variety of methodologies. The chosen methodology on the development side has an impact on the work of the software product manager and the interface between SPM and Development – in particular how requirements are handed over for implementation and how acceptance of project deliverables is managed. In reality most companies use a mix of different methodologies, be it agile, iterative, or stage-gate, often called waterfall. Popular agile and lean methodologies include Scrum, Kanban, and XP (eXtreme Programming).

At a high level, agile as well as iterative development are driven with small, controllable steps or iterations. With most agile methodologies, every iteration consists of analysis, design, coding and testing. With iterative only coding and testing are done in iterations while analysis and design are done upfront. With stage-gate, or waterfall, there are no iterations, but one stage or phase is done after the other. So with stage-gate and iterative, requirements are handled early whereas with agile requirements usually need to be handled in each iteration. There has to be a role in agile projects that deals with requirements. In Scrum that role is the Product Owner. In smaller agile projects, the SPM may assume the Product Owner role, but that does not scale up. So in larger projects, the Product Owner role needs to be filled with additional team members who cooperate tightly with the SPM. With agile methodologies, a larger part of requirements analysis is typically done within the development project than with more traditional methodologies.

With agile methodology and continuous integration (and even continuous deployment) new functionality can be delivered on a daily basis if customers are able to deal with such a high release frequency. In such environments release planning does not focus on each individual release, but rather on the requirements that are released within a certain time frame. The SPM challenge is to focus on the strategic and important items in such a high-pace environment.

Literature: *J. Bosch (2009), H.-B. Kittlaus (2012), D. Leffingwell (2011), R. Mironov (2008)*

EU4 Strategic Management

Duration: 2:00 h

Educational Objectives:

- EO4.1 Understand the essentials of strategic management.
- EO4.2 Know the activities that belong to Strategic Management.
- EO4.3 Understand the role of a product manager in relation to these activities.

Strategic Management Essentials

Strategic Management is an activity within an organization with the content to define, plan, agree, implement and evaluate the organization's strategy. It is part of the responsibility of executive management who can delegate preparatory work to staff functions. Strategic Management includes a number of elements related to software product management which are listed below (see also ISPMA SPM Framework in Fig. 1). Software product managers are typically not responsible for any of these activities, but they either participate in them, e.g. portfolio management, provide inputs, or make use of their outputs, e.g. product analysis.

Literature: W. Chan Kim, R. Mauborgne (2015); H. Mintzberg, B. Ahlstrand, J. Lampel (2008); M.E. Porter (1998)

Corporate Strategy

The strategy on the corporate level typically considers a timeframe of up to five years, or longer depending on domain. It consists of vision, mission, values and goals, corporate positioning, business model and financial plan, product portfolio and its evolution, resource and competency evolution, technology trends and innovation strategy, market trends and competitive strategy, policies and governance. Software product managers will have to provide input whenever the corporate strategy is revisited, and ensure that product strategies stay consistent with corporate strategy.

Literature: W.C. Kim, R. Mauborgne (2015); H.-B. Kittlaus, P. Clough (2009, p. 54 ff); M.E. Porter (1998)

Portfolio Management

Portfolio Management is an approach to define the investment strategy with regard to the products the company intends to offer in the strategic timeframe. Software product managers are typically asked to participate in the update cycles of the product portfolio by representing their products and providing input like roadmaps, forecasts, and investment requirements. The results of such an update cycle can have significant consequences for the individual product strategy.

Literature: R.G. Cooper, S.J. Edgett, E.J. Kleinschmidt (2001); H.-B. Kittlaus, P. Clough (2009, p. 47 ff)

Innovation Management

On the corporate level, innovation management needs to be aligned with the corresponding elements of the corporate strategy on a continuous basis. Alignment can be required in both directions: When innovation management leads to significant results these need to be incorporated into the corporate strategy and impacted product strategies in order to transform them into competitive advantage. On the other hand, changes of the corporate strategy need to be reflected in innovation management in order to shift resources according to the strategic directions. Software product managers need to ensure that their products benefit from innovation initiatives. The benefits of the open innovation model and crowd innovation, most commonly realized through the open source software model, should be carefully considered on the corporate level and aligned with the innovation management needs and the corporate strategies. That includes discussions about governance activities, contribution strategy, the degree of openness of the company and the identification of key differentiators for achieving or maintaining competitive advantage.

Literature: *H.W. Chesbrough (2005); T. Gorschek, S. Fricker, K. Palm, S. Kunsman (2010)*

Resource Management

On the corporate level, resource management needs to ensure that resources are available in the required quantities and qualities and at the required points in time so that the company is enabled to implement the corporate strategy and the aligned product strategies. This applies in particular to human resources, both in terms of numbers and skills. A software product manager needs to ensure that the resource requirements that result from the product strategy and plan can be fulfilled, i.e. are aligned with corporate resource management.

Market Analysis

It is of utmost importance for a corporation to have deep insight into trends and developments in the markets it wants to play in, and into the competitive landscape including competitors' strategies. The same holds true on the product level where the product manager needs reliable information. Larger corporations often have specialized market research departments that act as internal service units for the product managers. They conduct their own research and/or collect and evaluate information from market research agencies. In smaller companies the product manager may be required to do this.

Product Analysis

A product manager needs to be able at any time to know where his/her product stands compared to plan in order to take action if needed. This relates to cost vs. budget, revenue vs. forecast, number of licenses ordered (or installed) vs. plan and vs. market, and other measures that are considered relevant for the product. In most companies these numbers are provided by central corporate units like Finance, but the product manager needs to ensure that reliable numbers are available on a frequent basis.

Literature: *H.-B. Kittlaus, P. Clough (2009, p. 47 ff)*

EU5 Orchestration of the Organization's Units

Duration: 3:15 h

Educational Objectives:

- EO5.1.1 Know the core tasks of the functional area Development.
- EO5.1.2 Understand the product manager's role and responsibilities in relation to Development.
- EO5.2.1 Know the core tasks of the functional area Marketing.
- EO5.2.2 Understand the product manager's role and responsibilities in relation to Marketing.
- EO5.3.1 Know the core tasks of the functional area Sales and Distribution.
- EO5.3.2 Understand the product manager's role and responsibilities in relation to Sales and Distribution.
- EO5.4.1 Know the core tasks of the functional area Service and Support.
- EO5.4.2 Understand the product manager's role and responsibilities in relation to Service and Support.
- EO5.5.1 Know the product manager's role and responsibilities with regards to orchestration.
- EO5.5.2 Understand types of orchestration challenges and strategies.

The division of work in a company developing software products usually implies that there are separate functional areas for development, marketing, sales, consulting and other services, and support each of which has its own responsibilities, tasks, objectives, and often its own culture. The core tasks of the functional areas are listed in the respective columns in the ISPMA SPM Framework (Fig. 1). Neither the framework nor the following descriptions give recommendations or make assumptions about the mapping of these functional units to a company's organizational structure.

5.1 Development

Duration: 0:45 h

The development unit is responsible for all technical software aspects including the implementation of changes and extensions to the software.

Engineering management addresses all development aspects that are relevant across and above development projects. These include governance of product architecture, development processes and tools, configuration management, knowledge management, resource and skills management, development sourcing, and estimations.

Project management addresses the execution of a release plan in software development. This is typically done based on a project organization. For a project, the appropriate development methodology has to be selected and followed. This choice has an impact on the way SPM and the development project cooperate (see section 3.5). The project is usually responsible for writing internal documentation and contributing to the software-related external documentation.

Project requirements engineering is part of the project responsibility and follows a process similar to the product requirements engineering process. Once the contents of a release are defined, the

corresponding product requirements are transferred into project requirements and further refined. Project requirements can also address development-internal needs that are raised during the course of a project. Synchronization and tracking with product requirements is continuously required. With agile methodologies like Scrum a larger part of the requirements analysis is typically done in the development project.

User experience (UX) design addresses every aspect of the users' interactions with a software product or component with the purpose of shaping the user's behaviors, attitudes, and emotions about that product or component. Emotions include delight and annoyance about the product, excitement and fear in games, and a feeling of being in control when using decision-support software. UX design must take the human-system interaction processes, user interfaces, device and workplace ergonomics, service and content offered by the product, the context of product use and standards like ISO 9241 into consideration. UX design is a broader term covering or interacting with disciplines like graphic design, information architecture, Human-Computer-Interface (HCI) design, interaction design and usability engineering. Industry standards and user expectations for the user experience can be very different across types of software and market segments. The UX design scope and expectation for a product is described in the Product Strategy under Product Positioning.

Quality Management addresses the technical quality of software. It includes test concepts and infrastructure, technical support concepts and structure (together with the Support unit), a historical quality database, quality forecasting, and the execution of tests.

The product manager's orchestration responsibility includes the acceptance of results based on tests, agreements on release scope, schedules and estimates, tracking of the execution of plans, tracking of project vs. product requirements, negotiations and adjustments of plans including scope changes (if needed).

Literature: M. Hassenzahl, N.Tractinsky (2006); *ISO/AWI TR 9241-1 (1998)*; H.-B. Kittlaus (2012); H.-B. Kittlaus, P. Clough (2009, p. 94 ff); N. Lazzaro (2004); P. Le Callet e.a. (2013); Shneiderman e.a. (2013)

5.2 Marketing

Duration: 0:30 h

The marketing unit is responsible for all aspects in preparation and support of the product sales activities of a company including creation of product awareness and positioning of the product in the market. The actual split of responsibilities between Marketing and Sales may differ from company to company. In some companies, SPM and Marketing are combined.

Marketing planning addresses the development and negotiation of plans for all marketing-related activities during a given timeframe, often a year, including respective budgets. The plans can be product-specific, or for groups of products. They need to be synchronized with corporate and product strategies and plans, and the sales plan.

Customer analysis means the frequent analysis of existing or potential customers or groups of customers with regard to additional business opportunities and retention.

Opportunity management means the continuous pursuance of identified business opportunities with the objective to turn these opportunities into concrete product success. This may include the formulation of product requirements, development and implementation of new marketing approaches, and tight cooperation with Sales.

Marketing mix optimization means the selection, implementation and management of channels appropriate for a product, and the management of marketing partners within the product ecosystem. Both require tight cooperation with SPM and synchronization with the corresponding product strategy.

Product launches mean the introduction of a new product, version or release to the market. Marketing needs to orchestrate all activities that serve to create attention of existing or potential future customers, in the trade press, with market research agencies, and so on. Typically, SPM, Development, Sales, executives, partners, and sometimes customers are involved.

Operational marketing means the execution of the marketing plan, tracking of the relevant measurements, and taking corrective actions when measurements deviate from plan.

The product manager's orchestration responsibility includes the positioning of a product in the marketing plan, tracking of plan execution, cooperation in product launches, channel and partner management. The product manager may decide to selectively participate in marketing events.

Literature: *H.-B. Kittlaus, P. Clough (2009, p. 96 ff)*

5.3 Sales and Distribution

Duration: 0:30 h

The sales unit is responsible for all sales activities of a company. The actual split of responsibilities between Marketing and Sales may differ from company to company. Distribution means making the product available to the customer for use. This can include logistics to transport shrink-wrapped products to retail outlets, downloads on the internet, or delivery of software on physical storage media to a customer after acquisition. Manufacturing is usually not an issue for pure software products or services except for shrink-wrapped software. Distribution can be under Sales' responsibility, or with a central fulfillment unit.

Sales planning addresses the development and negotiation of plans for all sales-related activities during a given timeframe, often a year, including target values and incentives. The plans can be product-specific, or for groups of products. They need to be synchronized with corporate and product strategies and plans, and the Marketing plan.

Channel preparation means that the selected channels are enabled in time to sell a new product, version or release. It includes skills management and the provision of materials, web site, customer reference stories etc.

Customer relationship management means the systematic management of a company's interactions with customers, clients, and sales prospects. This includes customer communication, knowledge management, and customer requirements engineering. CRM must not only focus on short-term sales success, but also on long-term customer relationships.

Operational sales means the execution of the Sales Plan, tracking of the relevant measurements, and taking corrective actions when measurements deviate from plan. This includes offers and the negotiation of contracts, and the management of offers and contracts.

Operational distribution means ensuring smooth order and distribution processes, sufficient supply (in case of physical distribution), stable and easy online order and distribution, and smooth and correct billing/payment.

The product manager's orchestration responsibility includes the positioning of a product in the sales plan, in particular the incentives plan, tracking of plan execution, involvement in product-specific commitments to customers, in the handling of customer requirements (short-term sales vs. longer-term product perspective) as an input to product requirements, in deviations from standard terms and conditions, and in deviations from minimum price levels or price structure. The product manager may decide to selectively be involved in some pre-sales meetings with key customers.

Literature: *H.-B. Kittlaus, P. Clough (2009, p. 104 ff)*

5.4 Service and Support

Duration: 0:30 h

The service unit is responsible for all services offered and provided to customers. This includes both product-related services and consulting services that are not product-related (and therefore out of SPM scope). Product-related services can include education, installation, customization, operations, and user help desk covering technical and non-technical problems. Typically, these services are priced separately, sometimes they are bundled with software product offerings. Support units provide technical support to customers usually covered by maintenance contracts, and internal support for marketing and sales.

Service planning and preparation address the development and negotiation of plans for all product service-related activities during a given timeframe, often a year, including target values and incentives. The plans need to be synchronized with product strategies and plans, and the marketing plan. Preparation includes development of a technical basis (if required), forecast of demand (with SPM), resource management, skills development, and development of marketing material (with Marketing).

Service provisioning means the execution of the service plan, tracking of the relevant measurements, and taking corrective actions when measurements deviate from plan.

Technical support means the fulfillment of maintenance contracts. The typical support structure is:

- Level 1: Help Desk
- Level 2: Technical Maintenance
- Level 3: Change Team (typically in or with Development)

Customer calls need to be categorized in defects, requirements, and non-technical problems, and documented in a customer issue database.

Product-related documentation is required both internally and externally. Internal documentation includes documents like specifications, technical manuals etc. This documentation is usually developed by Development, and it is not intended for customers. External documentation refers to all documentation intended for use by people outside of the product-owning company such as end users, ecosystem players or service partners. For end users' guidance printed and/or online manuals, help functions, or step-by-step instructions need to be specifically developed and require collaboration between UX design, software development, technical support, and marketing. This work needs to be orchestrated by the product manager.

Marketing support means providing help to Marketing. It may include the production and distribution of marketing material, organization and execution of marketing events (conferences, user groups etc.), and documentation and tracking of marketing activities and their results.

Sales support means providing help to sales representatives and channels. It may include sales-related customer inbound and outbound communication, typically through a call center, organization and execution of sales events (customer visits, meetings etc.), documentation and tracking of sales activities and their results, provision of marketing material on request.

The product manager's orchestration responsibility includes the management of product-related services as part of the product offering, tracking of service execution, resource management and skills development.

Literature: *H.-B. Kittlaus, P. Clough (2009, p. 112 ff)*

5.5 The Role of SPM as Orchestrator

Duration: 1:00 h

One of the central tasks of SPM is to optimize the cooperation of all other units with respect to product-related goals. Conflicts of interest are bound to exist among the units and sometimes even within the product management organization. This task calls for successful negotiations in case of conflicts of interest.

The orchestration responsibility does not only apply to the functional areas discussed in 5.1 to 5.4, but also other areas, in particular the SPM units of other products within the company, Finance, Controlling, Human Resources, and Research. Here the product manager's orchestration responsibility includes the alignment of product strategies and plans, research and innovation initiatives for both functional and technical innovations, resource management, and availability of correct and timely measurements. When going international the product manager must take care that resources are provided for localization activities for all target countries. Often localization is done in the target countries. The product manager must then orchestrate timely execution of localization.

Typical orchestration challenges

- Between product managers of a product family: resource allocation, schedule dependencies.
- Between sales and development units: short-term goals and individual customers versus long-term abstract perspectives.
- Between product managers and development unit: release scope and dates, resource allocation and project planning (overall project portfolio, not details in each projects)
- Between product manager and marketing unit: brand marketing versus product marketing
- Between product manager and sales and distribution unit: discounting; product managers asking for reliable commitments regarding the sales volume of the product, sales only providing sales numbers for larger product groups, but not for individual products; what can be delivered realistically versus everything that customers want.

Strategies for avoiding/resolving conflicts

- Ensure a clear definition and separations of concerns, responsibilities and competencies, as well as clear definition of accountability
- Ensure a delineation of responsibility between product managers (mainly working in problem space) and project managers (mainly working in solution space)
- Define release planning, requirements management and quality assurance processes
- Separate contract giver (product manager) and contract taker (development unit)
- Establish clear channels of communication between product manager and different organizational parts (for example sales)
- Define contracting offers and agreements for system integration and outsourcing
- Define outsourcing (and subcontractor) policies to assure not only adequate deliveries, but also assuring retained knowledge about outsourced parts in-house, joint operational and quality control, and clear separation of responsibilities and accountability for deliverables

Literature: *S. Fricker, T. Gorschek, C. Byman, A. Schmidle (2010); H.-B. Kittlaus, P. Clough (2009, p. 42 f, p. 194 ff); H. Raiffa, J. Richardson, D. Metcalfe (2007); L. Thompson (2014)*

A. Addendum: SPM in Different Business Scenarios

As described in the preface the syllabus is business context agnostic and therefore applies to a wide range of industries. However, there are some specific considerations in certain business contexts. This addendum to the foundation level syllabus will outline some of them. These business scenarios are considered:

- Pure software products
- Software in software-intensive systems (embedded software)
- Software in software-intensive services (e.g. internet platforms or SaaS)
- Software managed by Corporate IT organizations (for one or multiple internal customers)

For each of these business contexts, we identify whether a chapter deserves specific consideration or is of minor relevance.

A.1 Pure Software Products

In this scenario the full contents of the syllabus is applicable. There are no specific considerations to be applied.

A.2 Software in Software-Intensive Systems

The amount of software embedded in components and systems is increasing at a high pace. In situations where the software contributes a significant part to the value proposition of the entire system, it ought to be managed from a software product management perspective.

Syllabus chapters with less relevance

Chapter 3.4 Product life cycle management: Profit considerations and market share are typically not relevant from the embedded software perspective.

Syllabus chapters with modified content

Chapter 2.1 Product Strategy Essentials

Strategy aspects are managed on the product level. Therefore, in the case of embedded software, the software product manager will focus on positioning with regard to the other components of the product; on the scope of the software; and on make or buy decisions. The other topics listed in

chapter 2.1 will be handled by the product manager responsible for the product in which the software is embedded.

Chapter 2.2 Business Aspects

The business aspects are usually managed on the product level. Therefore, the software product manager for embedded systems will typically either focus on the business aspects directly related to the software part, e.g. business cases and costing, while the overall business responsibility is with the product manager responsible for the product in which the software is embedded. Alternatively, the software product manager will be a part of a group of product managers doing the equivalent work for the entire product offering.

A.3 Software in Software-Intensive Services

The amount of software used in services like SaaS (Software-as-a-Service) or for internet platforms like community, communication, shopping, or contents platforms is increasing at a high pace. In all these instances the software contributes a significant part to the value proposition of the entire service. Therefore it ought to be managed from a software product management perspective.

Syllabus chapters with modified content

Chapter 2.1 Product Strategy Essentials

Strategy aspects are managed on the service level. If the software product manager role is separate from the role of service product manager the software product manager will focus on positioning with regard to the other components of the service; on the scope of the software; and on make or buy decisions.

Chapter 2.2 Business Aspects

The business aspects are usually managed on the service level. If the software product manager role is separate from the role of service product manager the software product manager will typically focus on the business aspects directly related to the software part, e.g. business cases and costing.

Chapter 2.4 Legal aspects

For SaaS offerings or other software-intensive services, the service provider needs license contracts with licensors of the software that is used as part of the service offering which explicitly allow this kind of use. Customers of such a service do not need license contracts, but only service contracts with the service provider.

Chapter 3.4 Product life cycle management:

Profit considerations and market share are typically considered from the service perspective.

A.4 Software Managed by Corporate IT Organizations

A growing number of corporate IT organizations in all industries is adopting software product management. The software components in an enterprise architecture tend to have very long life cycles which require a strategic view and continuity in management. Both cannot be ensured in a pure project organization. Some corporate IT organizations have been transformed into profit centers and may have multiple customers inside and outside of the corporation for the same software components which makes their business model very similar to a software vendor.

Syllabus chapters with less relevance

Chapter 2.2 Business Aspects

The relevance of the business aspects depends on how the business relationship between the corporate IT organization and the companies, business units or departments in the corporation is designed. If the corporate IT organization is run as a cost center its focus will be on business cases and costing, while the overall business responsibility is on the business side. The more the corporate IT organization is run as a business unit of its own, the more relevant the other business aspects become.

Chapter 2.3 Ecosystem Management

The ecosystem is usually restricted to the technology side, i.e. software and other technology providers and software development partners.

Chapter 3.4 Product life cycle management

Profit considerations and market share are typically not relevant unless the IT organization is run as a profit center and/or the corporation allows its business units to work with external competitors.

EU 4 Strategic Management

The relevance of the strategic management aspects depends on how the business relationship between the corporate IT organization and the companies, business units or departments in the corporation is designed. If the corporate IT organization is run as a cost center its focus will be on innovation management and resource management. The more the corporate IT organization is run as a business unit of its own, the more relevant the other strategic management aspects become.

Chapter 5.2 Marketing

A marketing organization typically only exists when the corporation allows its business units to work with external competitors. Only in these cases is this chapter relevant.

Chapter 5.3 Sales and Distribution

Customer relationship management is the only applicable area in this chapter. Customers are the company-internal departments and users.

Chapter 5.4 Services and Support

Marketing and sales support are not relevant.

Chapter 5.5 The Role of SPM as Orchestrator

All orchestration aspects regarding marketing or sales are typically not applicable.

Syllabus chapters with modified content

Chapter 2.1 Product Strategy Essentials

Strategy aspects need to be managed in close cooperation between the corporate IT organization and the companies, business units or departments in the corporation that are the customers of the software product. A most important aspect is the positioning of the software product in the enterprise architecture of the corporation over time.

Chapter 2.2 Business Aspects

Since a corporate IT organization usually has the responsibility for Operations, i.e. the run-time production environment, risk management has the added focus on operational risks.

Chapter 2.4 Legal Aspects

While contractual issues between the corporate IT organization and the companies, business units or departments in the corporation that are the customers of the software product are typically not relevant on a product level, contracts with software providers need special attention. Dependent on the industry the corporation, company or business unit is doing business in there may be specific legal or regulatory requirements.

EU 3 Product Planning

Product planning needs to be managed in close cooperation between the corporate IT organization and the companies, business units or departments in the corporation that are the customers of the software product.

Chapter 5.4 Services and Support

In a corporate IT organization, Services usually include Operations, i.e. the run-time production environment which is governed by the IT service management processes. The product manager is in a monitoring role and may be directly involved in critical situations.

Bibliography

Recommended SPM Overviews:

C. Ebert (2007): *"The Impacts of Software Product Management"*, *Journal of Systems and Software* 80:850-861.

S. Fricker (2012): *"Software Product Management"*, in A. Mädche, A. Botzenhardt, L. Neer (eds.): *Software for People - Fundamentals, Trends, and Best Practices*. Springer.

H.-B. Kittlaus, P. Clough (2009): *Software Product Management and Pricing – Key Success Factors for Software Organizations*. Springer.

I. van de Weerd, S. Brinkkemper, R. Nieuwenhuis, J. Versendaal, L. Bijlsma (2006): *"On the Creation of a Reference Framework for Software Product Management: Validation and Tool Support"*, *International Workshop on Software Product Management (IWSPM 2006)*.

Literature on specific aspects of SPM:

This literature has been used by ISPMA as the scientific basis for this syllabus. It is not required reading for course participants.

C. Anderson, C. Zeithaml (1984): *"Stage of the Product Life Cycle, Business Strategy, and Business Performance"*, *Academy of Management Journal* 27(1):5-24.

P. Berander, A. Andrews (2005): *"Requirements Prioritization"*. In A. Aurum, C. Wohlin (eds.): *Engineering and Managing Software Requirements*. Springer.

R. Berntsson-Svensson, T. Gorschek, B. Regnell, R. Torkar, A. Shahrokni, R. Feldt (2011): *Quality Requirements in Industrial Practice – an extended interview study at eleven companies*, *IEEE Transactions on Software Engineering*, vol. 99.

J. Bosch (2009): *From Software Product Lines to Software Ecosystems*, *13th International Software Product Line Conference (SPLC 2009)*, San Francisco.

P. Carlshamre, K. Sandahl, M. Lindvall, B. Regnell, J. Natt och Dag (2001): *"An Industrial Survey of Requirements Interdependencies in Software Product Release Planning"*, *5th International Symposium on Requirements Engineering (RE'01)*.

S. Chen, A. Cheng, K. Mehta (2013): *"A Review of Telemedicine Business Models"*. *Telemedicine and e-Health* 19(4):287-297.

H.W. Chesbrough (2005): *Open Innovation: The New Imperative for Creating And Profiting from Technology*. Harvard Business Review Press.

M. Cohn (2006): *Agile Estimating and Planning*. Prentice Hall PTR.

R.G. Cooper, S.J. Edgett, E.J. Kleinschmidt (2001): *Portfolio Management for New Products*, 2nd edition. Perseus Books.

- S. Fricker, S. Schumacher (2012): "Release Planning with Feature Trees: Industrial Case", 18th Intl. Working Conference on Requirements Engineering: Foundation for Software Quality (RefsQ 2012), Essen, Germany.*
- S. Fricker, T. Gorschek, C. Byman, A. Schmidle (2010): "Handshaking with Implementation Proposals: Negotiating Requirements Understanding", IEEE Software 27(2):72-80.*
- T. Gorschek, S. Fricker, K. Palm, S. Kunsman (2010): "A Lightweight Innovation Process for Software-Intensive Product Development", IEEE Software 27(1):37-45.*
- T. Gorschek, C. Wohlin (2006): "Requirements Abstraction Model", Requirements Engineering 11:79-101.*
- M. Hassenzahl, N. Tractinsky (2006): User Experience – A Research Agenda. Behavior & Information Technology 25(2):91-97.*
- M. Iansiti, R. Levien (2004): The Keystone Advantage – What the New Dynamics of Business Ecosystems Mean for Strategy, Innovation, and Sustainability. Harvard Business School Press.*
- ISO/AWI TR 9241-1 (1998): Ergonomics of Human-System Interaction. Part 1: Introduction to the ISO 9241 Series.*
- S. Jansen, M. Cusumano, S. Brinkkemper (2013): Software Ecosystems: Analyzing and Managing Business Networks in the Software Industry. Edward Elgar Publishers*
- M. Khurum, A. Khurum, T. Gorschek (2007): A Model for Early Requirements Triage and Selection (MERTS) Utilizing Product Line Strategies, 11th International Software Product Line Conference, Kyoto, pp. 97-104.*
- W.C. Kim, R. Mauborgne (2015): Blue Ocean Strategy: How to Create Uncontested Market Space and Make the Competition Irrelevant. Extended Edition. Harvard Business School Press.*
- H.-B. Kittlaus (2012): Software Product Management and Agile Development, in A. Mädche, A. Botzenhardt, L. Neer (eds.): Software for People - Fundamentals, Trends, and Best Practices. Springer.*
- B. Klemens (2006): Math You Can't Use – Patents, Copyright, and Software, Brookings.*
- R. Kostoff, R. Schaller (2001): "Science and Technology Roadmaps", IEEE Transactions on Engineering Management 48(2):132-143.*
- N. Lazzaro (2004): Why We Play Games: Four Keys to More Emotion Without Story. Report. XEO Design.*
- P. Le Callet, S. Böller, A. Perkis et al (2013): Qualinet White Paper on Definitions of Quality of Experience. White Paper. European Network on Quality of Experience in Multimedia Systems and Services.*
- D. Leffingwell (2011): Agile Software Requirements, Addison-Wesley.*
- L. Lehtola, M. Kauppinen, J. Vähäniitty (2007): Strengthening the link from business decisions to requirements engineering: Long-term product planning in software product companies, Proceedings of the 15th IEEE International Requirements Engineering Conference (RE'07), New Delhi.*

- M. McGrath (2000): *Product Strategy for High-Technology Companies – Accelerating Your Business to Web Speed*, 2nd edition. McGraw-Hill.
- D.G. Messerschmitt, C. Szyperski (2003): *Software Ecosystem – Understanding an Indispensable Technology and Industry*, MIT Press.
- H. Mintzberg, B. Ahlstrand, J. Lampel (2008): *Strategy Safari: The Complete guide through the wilds of strategic management*, 2nd edition. Financial Times Prentice Hall.
- G.A. Moore (2014): *Crossing the Chasm*, 3rd edition. Harper Business
- T.T. Nagle, J.E. Hogan, J. Zale (2014): *The Strategy and Tactics of Pricing – A Guide to Growing More Profitably*, 5th revised edition, Pearson Prentice Hall.
- T.T. Nagle, J.E. Hogan (2005): *What Is Strategic Pricing?*, in *Strategic Pricing Group Insights* (Monitor Group).
- A. Osterwalder, Y. Pigneur (2010): *Business Model Generation*. Wiley.
- R. Phaal, C. Farrukh, D. Probert (2007): "Strategic Roadmapping: A Workshop-based Approach for Identifying and Exploring Strategic Issues and Opportunities", *Engineering Management Journal* 19(1):3-12.
- R. Phaal, C. Farrukh, D. Probert (2004): "Technology Roadmapping – A Planning Framework for Evolution and Revolution", *Technological Forecasting and Social Change* 71:5-26.
- K. Pohl, K. Rupp (2011): *Requirements Engineering Fundamentals*. Rocky Nook Computing.
- K.M. Popp, R. Meyer (2010): *Profit from Software Ecosystems: Business Models, Ecosystems and Partnerships in the Software Industry*, Books on Demand.
- M.E. Porter (1998): *Competitive Strategy*, Free Press.
- H. Raiffa, J. Richardson, D. Metcalfe (2007): *Negotiation Analysis – The Science and Art of Collaborative Decision Making*. The Belknap Press of Harvard University Press.
- V. Rajlich, K. Bennett (2000): "A Staged Model for the Software Life Cycle", *IEEE Computer* 33(7):66-71.
- B. Regnell, S. Brinkkemper (2005): "Market-Driven Requirements Engineering for Software Products". In A. Aurum, C. Wohlin (eds.): *Engineering and Managing Software Requirements*. Springer.
- G. Ruhe (2010): *Product Release Planning: Methods, Tools and Applications*, CRC Press.
- M. Schmidt (2002): *The Business Case Guide*. Solution Matrix.
- B. Shneiderman, C. Plaisant, M. Cohen, S. Jacobs (2013): *Designing the User Interface: Strategies for Effective Human-Computer Interaction* (5th revised edition). Addison-Wesley
- M.N. Sodhi, N.N. Sodhi (2007): *Six Sigma Pricing: Improving Pricing Operations to Increase Profit*, Pearson FT Press.
- L. Thompson (2014): *The Mind and Heart of the Negotiator* (6th edition). Pearson Prentice Hall.

C. Wohlin, A. Aurum (2005): "What is Important when Deciding to Include a Software Requirement in a Project or Release?", International Symposium on Empirical Software Engineering.

D. Zowghi, C. Coulin (2005): "Requirements Elicitation: A Survey of Techniques, Approaches, and Tools". In A. Aurum, C. Wohlin (eds.): Engineering and Managing Software Requirements. Springer.