



ISPMA

INTERNATIONAL
SOFTWARE PRODUCT MANAGEMENT
ASSOCIATION

Software Product Management Excellence in Orchestration

V.2.0 Student Edition

This syllabus was written by the following members of the International Software Product Management Association (ISPMA®): Hans-Bernd Kittlaus (editor), Garm Lucassen, Barbara Hoisl, Greg Prickril.

We thank all honorary authors and contributors.

Copyright © 2022 for this syllabus is with the authors mentioned above. The rights have been transferred to the International Software Product Management Association.

License to use: CC BY 4.0 (creativecommons.org/licenses/by/4.0/)

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 “ISPMA Software Product Manager - Excellence in Orchestration” is to deepen the understanding of the role of product managers about orchestration tasks.

The syllabus “SPM: Excellence in Orchestration” covers the key elements of software product management practices related to orchestration tasks according to the ISPMA® SPM Framework, which is well supported by research and industrial practice. The syllabus corresponds to **a two-day industry course**.

The syllabus addresses the needs of product managers who must orchestrate the activities of other organizational functions like development, sales, and marketing. The syllabus is the basis for examination to certify that the examinee has achieved the degree of knowledge described in this syllabus.

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).

The syllabus consists of **seven chapters**. Each chapter has educational objectives (EOs) that are enumerated following the chapter headers (EO1.1., EO1.2,...). Each chapter also includes the duration suggested to teach it. 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 three cognitive levels of knowledge phrased using the verbs “knowing” for level 1 and “understanding” for level 2, and “applying” for level 3. These three verbs are placeholders for the following:

- L1 (know): enumerate, characterize, recognize, and name.
- L2 (understand): reflect, analyze, justify, describe, judge, display, complete, explain, elucidate, elicit, formulate, identify, interpret, reason, translate, distinguish, compare, understand, suggest, and summarize.
- L3 (apply): perform, execute, develop, and adapt.

Each EO in the syllabus has one or more of these cognitive levels assigned to it.

ISPMA®’s Excellence syllabi are designed to put special focus on exercises that address L3 objectives. It is the trainer’s responsibility to select exercises and to define concrete realistic scenarios in which all the selected exercises can be performed by the participants. ISPMA® recommends spending about 50% of the available time on exercises. In trainers’ material, exercises are described in abstract terms. This syllabus also suggests role plays as a rich method for participants to internalize orchestration concepts.

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.

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 two-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 an appropriate, realistic scenario for the exercises described in this syllabus so that participants get an opportunity to apply the training contents. 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® certificate “ISPMA® Software Product Manager Excellence in Orchestration.” All chapters are relevant for the exam. 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 assume general knowledge of, and some experience in, the management or development of software products or software in software-intensive systems.

The educational 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 the Certificate “ISPMA® Certified Software Product Manager – The Foundation” or at least three years of experience in software product management. However, this is a generic recommendation and might not apply to all situations or course participants.

Terminology

The term SPM is used as an abbreviation for Software Product Management. It represents the function of SPM and not the individual person.

This curriculum usually uses a gender-neutral form. In cases where the masculine form is used, this is done for readability reasons and represents any other gender as well.

The terms used in this syllabus are consistent with the glossary of the ISPMA® available at ispma.org.

Table of Contents

EU1	Introduction	1:30 h
EU2	Leadership Skills for Software Product Managers	2:00 h
EU3	Orchestrating Development and User Experience Design	2:30 h
EU4	Orchestrating Marketing	1:30 h
EU5	Orchestrating Sales and Fulfillment	1:30 h
EU6	Orchestrating Delivery Services and Support	1:30 h
EU7	Complex Scenarios Involving Multiple Functions	1:30 h
	Bibliography	

EU1 Introduction

Duration: 1:30 h

Educational Objectives:

- EO1.1. Understand the responsibility of product managers to lead a cross-functional product team that typically does not report to her/him
- EO1.2 Understand that excellence in orchestration requires the SPM to drive improvement in relationships with other functions
- EO1.3 Understand why software product managers need to clarify roles and responsibilities across the cross-functional product team

Product managers are often expected to be „mini-CEOs“ for their products, „responsible for managing software with the objective to achieve sustainable success over the life cycle of the software product“ (*ISPMA SPM F (2022)*).

To achieve that, product managers collaborate with a wide range of stakeholders from both outside and inside their organization. Inside the organization, they work together with other business functions in a cross-functional product team that ideally stays in place for the long term, i.e., not merely for a single release. The purpose of this team is „... to manage all the elements needed to achieve the financial, market, and strategic objectives of the product as a business“ (*S. Haines (2021)*).

The cross-functional product team has members from different departments in the organization, and may often be virtual, with members from different geographic locations or cultural backgrounds. In this setup, product managers are challenged to achieve “influence without authority, accountability without control” (*K. Hall (2013), p.1*).

The ISPMA® SPM framework (see Figure 1) identifies four key functional areas that a software product manager usually needs to orchestrate: Development, Marketing, Sales & Fulfillment, and Delivery Services & Support. The ISPMA® SPM Framework (Fig. 1) lists the core tasks that each of these functions performs in the respective columns of the framework. When an activity in one of these four key areas is outsourced to an external partner, the product manager needs to orchestrate this partner as well. Orchestration of other areas is often required, “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.” (*ISPMA SPM F (2022)*).

Each of those functions has its own responsibilities, tasks, objectives, measures of success, and often its own culture. Therefore, conflicts may arise between the software product manager and other functions.

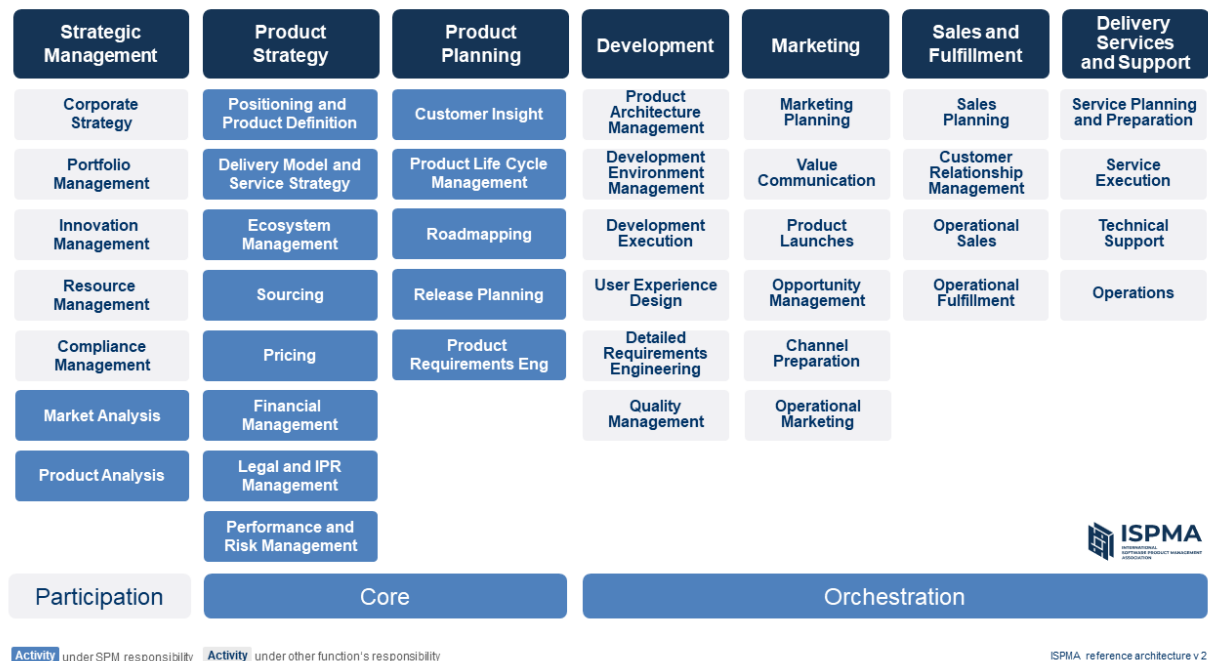


Fig. 1 SPM Framework V.2.0

To avoid or to resolve these conflicts, the following strategies can help (*ISPMA SPM F (2022)*):

- Clear definition and separations of concerns, processes, responsibilities and competencies, clear definition of accountability
- Delineation of responsibility between product managers and project managers
- Definition of release planning, requirements management, and quality assurance processes
- Establish clear channels of communication between product manager and different organizational parts (for example sales)

In addition to those strategies, roadmaps are also an important tool that supports collaboration with other functions.

To achieve **excellence** in orchestration, software product managers need to drive process improvement in working with other functions. Excellent software product managers are not just reacting to operational challenges. Instead, they pro-actively foster better relationships and drive better processes for collaborating with these other functions.

References: *ISPMA SPM F (2022), K. Hall (2013), S. Haines (2021), H.-B. Kittlaus (2022)*

EU2 Leadership Skills for Product Managers

Duration: 2:00 h

Educational Objectives:

- EO2.1 Apply the right mindset to succeed in orchestration
- EO2.1.1 Understand why orchestrating other functions requires a mindset similar to the mindset required by members of a matrix organization
- EO2.1.2 Understand the four mindsets of influential product management and their key characteristics
- EO2.2 Apply the “sources of power” concept to broaden your influence
- EO2.2.1 Know the definition of power and related concepts
- EO2.2.2 Understand the sources of power and their implications for leaders
- EO2.2.3 Apply techniques and approaches for increasing influence/power
- EO2.3 Apply conflict management techniques to resolve conflicts constructively
- EO2.3.1 Know common sources of conflict with all orchestration departments
- EO2.3.2 Understand how to create an escalation path process
- EO2.4 Apply best practices on negotiation
- EO2.4.1 Understand the difference between positions and interests
- EO2.4.2 Know the definition of BATNA
- EO2.4.3 Apply the concepts of interests vs. positions and BATNA to prepare for negotiations and to conduct negotiations

2.1 Mindset for Influencing without Authority

As was mentioned in EU1, product managers are challenged to achieve “influence without authority, accountability without control” (*K. Hall (2013), p.1*). This challenge is similar to the challenges faced by individuals working in a matrix organization. Therefore, software product managers can benefit from the mindset and specific leadership skills that are helpful for succeeding in a matrix organization.

K. Hall (2013) outlines that traditionally, organizations were vertically organized along functions or geography. In contrast, many large organizations today operate based on a matrix structure, where individuals report to multiple managers. For individuals in matrix organizations, this may lead to unclear or even conflicting roles and goals, and to an increase in uncertainty and conflict.

To successfully navigate these challenges, Hall claims a certain mindset is required: A successful matrix manager “... relishes the flexibility, autonomy, and breadth that the matrix gives them.” (p . 219). This mindset can also help software product managers succeed in orchestrating other functions.

Hall recommends that instead of belaboring the lack of authority and role clarity, successful matrix managers set their own commitments, based on what they see needs to be done. Then, they identify the stakeholders that need to be influenced to get the job done, and they actively seek out resources that are needed to meet their commitments. This style of leadership will increase their influence in the organization and earn them the right to take on more responsibility.

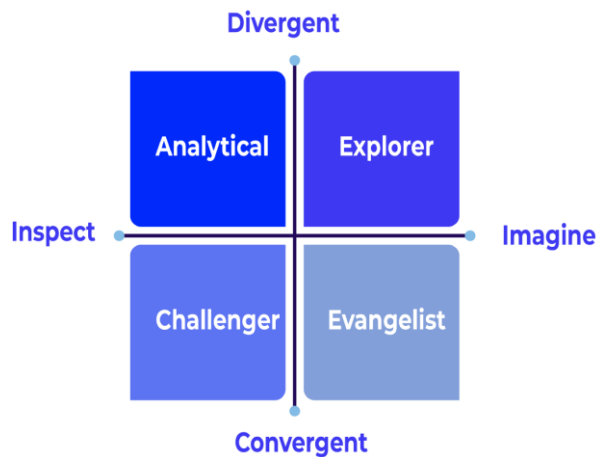


Fig. 2 Four mindsets of influential product management (K. Sandy (2020))

Sandy puts his focus directly on influential product management with his four mindsets that need to be applied in parallel depending on the situation:

- **Explorer:** expands the solution space based on creative thinking, e.g. design thinking
- **Analytical:** makes decisions based on data,
- **Challenger:** challenges assumptions and ideas to counter confirmation bias,
- **Evangelist:** get buy-in from stakeholders and team through effective communication.

References: K. Hall (2013), K. Sandy (2020), M. Lewrick, P. Link, L. Leifer (2020)

2.2 Sources of Power

Because the various company functions rarely report to the SPM organization, software product managers must find creative ways to influence to achieve product vision, goals, and objectives. Understanding what power is, where it comes from, and how to increase it can help software product managers increase their influence.

Social communication studies have theorized that leadership and power are closely linked. They have further suggested that some forms of power affect one's leadership and success. That idea is often used in organizational communication and throughout the workforce.

Bases of Power

Social psychologists John R. P. French and Bertram Raven released a study in 1959 which defines the term power and defines “bases” of power. In the French & Raven study, power is defined essentially as the ability to influence others. French and Raven defined the following “bases of power”:

Reward power	Reward power is based on a person being able to control the likelihood that another person will be rewarded. Close alignment of SPM with executive leadership may offer SPM some reward power.
Coercive power	Coercive power uses the threat of force to gain compliance from another with physical, social, emotional, political, or economic means. Typically, SPMs have very little coercive power over other functions as they have no direct management authority over them.
Legitimate power	Legitimate power comes from an elected, selected, or appointed position of authority and may be underpinned by social norms. An employee's direct manager often can exert legitimate power because the organization has authorized that manager to influence the people that report to him or her. SPMs typically have some legitimate power granted to them by the organization. Some development life cycle models, like Scrum, empower the product owner, who may report to the product manager, by giving the product owner ownership of the product backlog.
Referent power	Referent power is based on one person's strong identification with another. A person identifying with another person is likely to adopt attitudes and beliefs similar to that other person. Referent power is often the primary source of power for software product managers. Product managers can increase their referent power by defining and executing a compelling product strategy and investing in trustful relationships with other organizational functions.
Expert power	Expert power is based on one person's perception of another's knowledge in a given area. For example, a person's thinking and behavior regarding a legal case might easily be influenced by advice from a lawyer. SPMs often have expert power based on their knowledge of stakeholders, the business aspects of the software product, and the domain addressed by the software product.

Table 1: Bases of Power

Product managers typically exert the most influence using referent and expert power as their legitimate power is limited (depending on organizational setup) and they have little or no reward or coercive power relative to other organizational functions.

2.3 Managing Conflict

Product managers are expected to orchestrate across the organizational boundaries of an organization, without the power to direct other business functions. Consequently, conflict with other business functions will occur that prevents them from reaching their goals. Nevertheless, conflict is also beneficial for organizations (Rahim 2015). A certain level of disagreement necessitates two parties to negotiate to find a single, mutually satisfactory solution (Zartman & Rubin (2011)). This chapter introduces the basics of the *human-process* and *techno-structural* approaches to conflict resolution by means of intervention by a third party after *escalation*. (Rahim (2015)).

Human-process

The human-process approach “*attempts to improve organizational effectiveness by changing members’ attitudes and behavior regarding conflict*” (Rahim (2015)). The main approach to achieving this goal is to educate all stakeholders on the *five styles of handling interpersonal conflict*. Figure 3 and Table 2 briefly summarizes each style, differentiated on two dimensions: *concern for self* or how much a person attempts to satisfy his own concerns and *concern for others* or how much that same person wants to satisfy the concern of the other party or parties.

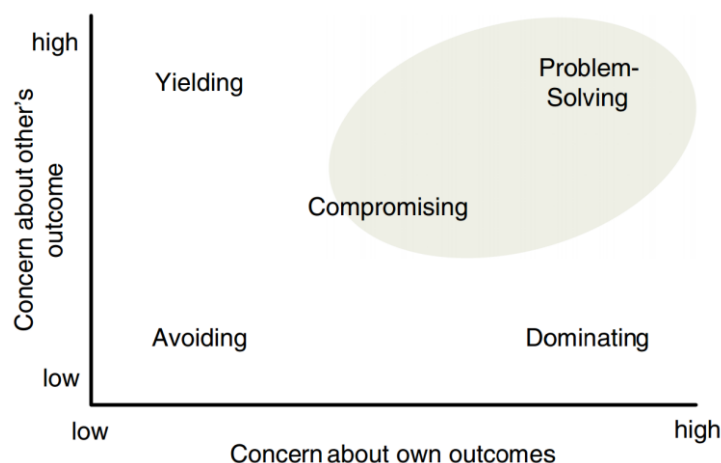


Fig. 3: Styles of handling interpersonal conflict, Fricker/Grünbacher (2008) based on Rahim (2015).
 Grey-shaded: desired, value-creating behaviors.

Problem-Solving high concern for self and others	A problem-solving approach leads to full collaboration between parties. All try to be open, exchange information, and examine differences to reach a mutually acceptable solution. This style has two distinctive elements: confrontation and problem solving: “confrontation involves open and direct communication which should make way for problem-solving. As a result, it may lead to creative solutions to problems.”
Yielding low concern for self and high concern for others	A party sacrifices his personal interests to satisfy the interests of the other party. The former reacts to a perceived hostile act of the latter with low hostility or even positive friendliness. During a negotiation, the former will attempt to play down the differences and emphasize commonalities between the two parties.
Dominating high concern for self and low concern for others	A competitive or dominating individual will do anything to achieve their personal objectives. Consequently, he or she ignores the needs and expectations of the other party. A dominating person tries to impose his will by sheer force, e.g. on subordinates and commands their obedience.
Avoiding low concern for self and others	A party avoids negotiation by “I see no evil, hear no evil, and speak no evil.” The party attempts to postpone an issue, withdraw from the situation, and refuse to acknowledge the conflict. When the conflict is avoided, the interests of neither party are satisfied.
Compromising intermediate in concern for self and others	Compromising is characterized as give-and-take or sharing and often used to conclude negotiations rapidly. Both parties give up something to make a mutually acceptable decision. They might split the difference, exchange concessions, or search for middle ground. This style is in the middle of all other styles.

Table 2: Five styles of handling interpersonal conflict. Paraphrased from Rahim (2015)

Read more about the five styles of handling interpersonal conflict and when it is appropriate to adopt specific styles in “*Managing conflict in organizations*” (Rahim (2015)).

Techno-structural

The techno-structural approach attempts to “*improve organizational effectiveness by changing the organization’s structural design characteristics*” or managing the amount of conflict by introducing organizational changes.

A variety of organizational change techniques exists that one can choose. In a Harvard Business Review article, Weiss and Hughes (2005) introduce a comprehensive set of best practices for *managing disagreements at the point of conflict* and *strategies for managing conflict upon escalation up the management chain*. The former empowers employees to resolve a conflict themselves. Sometimes, however, it is necessary to escalate to a superior who can decide based on the latter strategies.

Weiss and Hughes recommend implementing three strategies for each type to transform conflict from a major liability into a significant asset. We summarize these strategies below in Table 3 and Table 4.

Be aware that it depends on the corporate culture of an organization in which cases escalations are seen as acceptable.

Devise and implement a common method for resolving conflict
A company-wide process for resolving disagreements prevents useless debate about who's right and wrong or haggling over small concessions. A well-defined, well-designed conflict resolution method will reduce transaction costs and foster an environment in which innovative outcomes emerge from discussions. Unfortunately, no conflict resolution method is universally applicable. To be effective, design a conflict resolution process that offers a clear, step-by-step process and makes it an integral part of existing business activities. Processes that are only used in exceptional circumstances will be unsuccessful.
Provide people with criteria for making trade-offs
From time to time, two parties need to make zero-sum trade-offs between competing priorities – situations in which it is unclear which decision is best. Define the criteria for making such choices clearly by setting concrete ground rules for your organization. Salespeople that know that 5 points on a customer satisfaction scale are more important than 10 points of market share are now capable of making informed decisions. Even when the criteria are not decisive, they will foster productive communication around a common objective.
Use the escalation of conflict as an opportunity for coaching
Senior management should view each escalation of conflict as an opportunity to teach employees about how to resolve conflicts effectively. To mitigate people's natural tendency to let their bosses sort out disputes, the bosses should push their respective employees to consider <i>"the needs of the other division, alternatives that might best address the collective needs of the other division, and the standards to be applied in assessing the trade-offs between alternatives"</i> . While this approach requires more time from senior managers initially, over time it effectively reduces the time senior managers need to spend on resolving escalated conflicts.

Table 3: Three strategies for managing disagreements at the point of conflict

Establish and enforce a requirement of joint escalation
In a typical conflict, all parties will try to get support from their direct leadership leading to a vicious circle. Prevent this by enforcing people to present disagreements to their bosses in unison. Now the ultimate decision maker will have a balanced view of the perspectives on the conflict, its causes and knows all the possible solution. Moreover, the number of problems that are escalated to upper management decrease!
Ensure that managers resolve escalated conflicts directly with their counterparts
An unresolved dispute typically travels up the management chain until a senior manager with the most organizational clout makes a unilateral decision. This dynamic breeds organization-wide resentment in the form of “we’ll win next time!”, making friendly conflict resolution harder and harder. Moreover, unilateral decisions lead to inefficiency, ill feelings and bad decision-making. Instead, all managers should formally commit to dealing with escalated conflicts directly with their management counterparts in other departments.
Make the process for escalated conflict resolution transparent
Having resolved a conflict, managers at most companies announce the decision and move on. This rush prevents employees from learning how to resolve similar issues in the future. Instead, management should explain how a decision was reached and the specific aspects that were weighed. Do not share every single detail of the process, but an honest discussion of the relevant trade-offs informs and empowers people in future conflicts.

Table 4: Three strategies for managing conflict upon escalation

References: Fricker, S.A., Grünbacher, P. (2008); Rahim, M. A. (2015); Weiss, J., & Hughes, J. (2005); Zartman, I. W., & Rubin, J. Z. (2000)

2.4 Negotiation Skills

Negotiation is a discussion between multiple parties regarding the terms of an agreement. As a leader in the product organization responsible for aligning the requirements and efforts of multiple stakeholders, negotiating is a critical capability for most software product managers. Product managers frequently negotiate regarding the following topics (among others):

- Release scope and timing with executive leadership, development, and others
- Budget for other functions like marketing and sales
- Contracts with third parties like software suppliers or service providers
- Product pricing/discounting with individual customers

The Harvard Negotiation Project developed the concept of **principled negotiation** as described in their seminal work on negotiation, "Getting to Yes", to address what they considered non-productive negotiation approaches and tactics.

Principled Negotiation

Principled negotiation encourages negotiators to bargain over interests rather than position and defines four principles:

- Separate the people from the problem: By nature, people often conflate the relationship with the substance of the conflict. Discussion of each other's emotions, fears, and perceptions can lead to better understanding.
- Focus on interests, not positions: A position is what a party in a negotiation is willing or unwilling to accept. Interests represent the party's wants or needs relative to their position.
- Invent options for mutual gain: Creatively think of options that address negotiation parties' interests.
- Insist on using objective criteria that provide a means of resolving differences in interests outside the context of the negotiating parties' wills.

Principled Negotiation also defines the concept of a Best Alternative to a Negotiated Agreement (BATNA), i.e., the most advantageous alternative course of action a party can take if negotiations fail and an agreement cannot be reached. A party in a negotiation should typically not accept an outcome that is less desirable than their BATNA. It is important to not only consider your own BATNA when preparing for a negotiation, but also that of the other parties.

Former FBI negotiator Chris Voss suggests using negotiation tactics that take advantage of human psychology by exploiting well-known cognitive biases, for example the Framing Effect or Loss Aversion.

References: Fisher, R., Ury, W.L., Patton, B. (2012); Voss, C. (2016)

EU3 Orchestrating Development and User Experience Design

Duration: 2:30 h

Educational Objectives:

- EO3.1 Understand that the organizational setup, typical roles, and processes used in Development and UX Design vary considerably and that the approach to working with these functions needs to be tailored to their respective setup
- EO3.2 Apply the understanding of the inner workings of the Development and UX Design functions to establish effective processes for collaborating with Development
- EO3.3 Understand the impact of agile development, continuous delivery, and Devops on the software product manager
- EO3.4 Understand typical areas of conflict between Product Management versus Development and User Experience Design
- EO3.5 Apply the understanding of typical conflict areas to work proactively with Development and UX Design to prevent conflicts where possible, to surface hidden conflicts early, and to address them constructively by either resolving or escalating them

3.1 Introduction

“The development unit is responsible for all technical software aspects including the implementation of changes and extensions to the software.” (*ISPMA SPM F (2022)*). Key activities and areas of responsibility include product architecture management, development environment management, development execution, detailed requirements engineering, and quality management. These are described in more detail in the foundation syllabus (*ISPMA SPM F (2022)*).

The development function exercises a strong influence on the product’s functional capabilities and non-functional qualities, as well as the user experience. Therefore, successful collaboration with Development and User Experience (UX) Design is a key success factor for software product managers - and there are many possible areas for conflict. In the SPM Framework, UX Design is positioned in the Development column from a functional perspective. Organizationally, it can be unit of its own, or close to Software Product Management.

Software product managers need to work pro-actively with Development and UX Design to prevent conflicts where possible, to surface hidden conflicts early, and to address them constructively by either resolving or escalating them.

The setup and processes used in Development and User Experience Design vary widely, depending on the type of software product, the organization’s needs, market environment, development method, e.g. waterfall, iterative, or agile, and the process chain from code development to product delivery.

To successfully orchestrate development and User Experience Design, software product managers need to tailor their approach to collaboration to fit with the setup and processes used by these functions.

3.2 Product Architecture Management

The product architecture is designed, maintained and managed under the lead of the chief architect of a software product, sometimes supported by a team of architects. Product Architecture has a significant impact on a software product with regard to evolution and flexibility. It consists of a number of dimensions that are listed in the table below and needs close cooperation between product

manager and architect. This table suggests a split of responsibilities. The aspects that fall under the responsibility of the software product manager are part of the software product strategy.

Architecture dimension	Software Product Manager	Technical architect
Offering architecture	Lead – define separately priced components of the product (suite, platform) offering, and tailorability options.	Ensure technical feasibility including access management, support for licensing and pricing approach, etc.
Business architecture (only for application software)	Lead – define domain-specific architecture, i.e. a logical data model, process model, business object model, etc.	Ensure that technical architecture supports the implementation and change management of the business architecture.
Technical architecture	Define the relevant strategy elements such as the delivery model, pricing approach, release approach, quality scope, and IT stack constraints.	Lead – define the technical architecture in line with the business architecture and strategic and technical requirements, e.g. IT stack, programming languages, etc.
Tailorability architecture	Define the tailorability strategy as part of the delivery model in line with the ecosystem strategy and sales and marketing strategies.	Lead – Define the tailorability architecture as part of the technical architecture in line with the tailorability strategy.
Governance	Ensure that development activities remain consistent with the planned offering, business architecture, and compliance goals.	Lead – ensure that development teams implement in line with the defined technical architecture, and that the technical architecture is only changed based on a well-defined and controlled process.

Table 6: Software Architecture Dimensions

Business Architecture needs to be defined by modeling experts in cooperation with domain experts. With software vendors they may be part of the SPM organization, for corporate IT organizations they are usually on the business side.

The product architecture can serve as an enabler for competitive edge and market differentiation over time. One method to secure this, is to have a *defining technology* in the software product. These are core technology assets that may be used in multiple components, products and offerings, must be difficult to copy and are the basis for significant customer value in a way that creates sustainable differentiation. Both the product manager and the architects must pay attention to the continuous improvement and protection of the defining technology of their software product.

It is one of the hardest problems in creating and developing a software product to design an architecture that balances flexibility and cost of implementation. Changing requirements over the life cycle of a product may require changes to the architecture as well which in the worst case can mean a full re-implementation of the product code.

3.2 Continuous Everything and DevOps

Software organizations increasingly automate the processes that lead from finalized code changes to the deployment of new software in production environments.

The software product manager needs to be aware of the technical capabilities of their development organization and needs to make sure that actual delivery to customers is aligned with business needs: how to use these capabilities must be a business decision, based on what the market can absorb.

The processes that lead from finalized code changes to deployment in production environments include

- Continuous integration: automated integration, build, and test in the development environment
- Continuous delivery: automated push of software into the production environment or delivery to customers

If both continuous integration and continuous delivery capabilities are in place, the organization can choose how frequently to deliver code changes to customers. If every code change is automatically pushed to production, this is called continuous deployment. Continuous deployment can result in multiple production deployments per day, for example in a SaaS delivery model. (*Definitions according to Martin Fowler in IL1; see also J. Humble, D. Farley (2010)*).

To implement continuous delivery and continuous deployment, Development needs to collaborate closely with other functions. If the software is also operated by the software vendor, for example in a SaaS (Software as a Service) delivery model, tight integration with the Operations function is required.

This approach is called DevOps which is a development methodology aiming for a tighter cooperation between Development and Operations to achieve better quality of software products, shorter time to market, and improvements in operational efficiency. At the core of a DevOps setup is a collaborative culture. To strengthen this culture, product managers need to build a clear and common view for the product vision, strategy, and principles across the functional units involved. They can form the basis for daily decision making and execution. Once a DevOps approach with elements like automated testing and a seamless tool chain is established, product managers not only benefit from improved time-to-market, but also up-to-date product status insights that enable fact-based discussions with the functional units. “DevOps is about aligning the incentives of everybody involved in delivering software, with a particular emphasis on developers, testers, and operations personnel.” (*J. Humble, J. Molesky (2011)*)

In that context, it can be beneficial to establish a “release heartbeat” where new functionality is released to the market at regular intervals, e.g. once every week or every six months. The heartbeat drives alignment and efficiency within the software organization and properly sets expectations in the market.

3.3 Typical Areas of Conflict - Development

Typical areas of conflict between SPM and Development include:

- **Release planning:** Disagreement regarding the priority of requirements
 The disagreements may be related to individual requirements or features, or to classes of requirements: for example, SPMs often focus heavily on functional requirements, while Development may push to spend more effort on architectural improvements, code cleanups, reduction of technical debt, or for addressing non-functional requirements

- **Release planning:** Development “does what they want” instead of executing the release plan. For example, Development may consider the list of requirements to incorporate in next release or the planned release date as unrealistic in light of existing resource and schedule constraints - and simply goes off to execute based on priorities they set themselves.
- **Detailed Requirements Engineering:** the software product manager does not agree with the way the development function has interpreted and refined certain product requirements.
- **Agile Development – Sprint Planning:** over multiple short sprints the product increasingly evolves into a direction that is not in line with the release themes and epics defined by the product manager.
- **Continuous Deployment - Disagreement on Frequency of Deployments**
Once continuous deployment capabilities have been set up, Development and Operations may want to use it to full extent. However, the software product manager may want to exercise more control over what is deployed when, including the power to group changes together or to delay deployment of certain changes.
- **DevOps – Quality Problems from Imperfect Implementation**
DevOps requires a smooth automated process execution, from the time when the developer commits a piece of changed code to the code database until that code is actually deployed in the runtime environment. Quality problems can result from insufficient process design, technical problems during execution, insufficient coverage of automated testing etc.

3.4 User Experience (UX) Design

User Experience Design can be anchored at different positions in the org chart, but often resides within or close to the development function.

“User Experience (UX) design can be a key factor for differentiation and competitive strength. It 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.” It is much broader than just usability, as it is “... covering or interacting with disciplines like graphic design, information architecture, Human-Computer- Interface (HCI) design, interaction design and usability engineering.” (ISPMA FL 2021)

M. Cagan (2013) distinguishes four design-related activities that are critical to the success of software products: interaction design, visual design, rapid prototyping and usability testing. These four roles need to “... work closely with the (software) product manager to discover the blend of requirements and design that meet the needs of the user.” (*M. Cagan (2013)*).

Startups following the Lean Startup principles and using agile development practices have developed *Lean UX* to “... break the stalemate between the speed of Agile and the need for design in the product-development lifecycle.” (*J. Gotthelf, J. Seiden (2013)*).

Lean UX is based on three foundations: design thinking, agile software development, and the Lean Startup method. All three foundations emphasize experimentation, rapid iterations, and deep customer and user involvement. Key approaches and techniques of these underlying foundations are then applied to the design process. “... this is the essence of the Lean UX approach. Only design what you need. Deliver it quickly. Create enough customer contact to get meaningful feedback fast.” (*J. Gotthelf, J. Seiden (2013)*)

3.5 Typical Areas of Conflict - UX Design

Due to the objectives of UX Design, there is a significant overlap with the product manager role, especially in the following areas:

- Developing a deep understanding of customers' real needs
- Understanding intended product usage
- Developing product scope and product definition
- Eliciting high-level product requirements

In these areas, software product managers may find that UX designers are powerful allies that help them define a product that serves customers and users even better - or they might be in stark conflict, quarreling over decisions and accountabilities.

To avoid unnecessary conflicts, software product managers should be careful to articulate requirements on the problem level and to avoid prescribing design solutions. On the other hand, they need to ensure that UX designers accept their product vision and product definition.

3.6 SPM's Focus Areas for Orchestrating Development and UX Design

An SPM needs to focus on the following areas when orchestrating Development and UX Design:

- Product architecture that balances flexibility and cost of implementation
- Acceptance of results based on verification and validation tests
- Release scope and dates, planning of development execution
- Execution of plans
- Synchronization and tracking of detailed vs. product requirements
- Estimates
- Resource, knowledge, and skills management

References: M. Cagan (2013); J. Gotthelf, J. Seiden (2013); J. Humble, D. Farley (2010); J. Humble, J. Molesky (2011); ISPMA SPM F (2022); H.-B. Kittlaus (2022)

EU4 Orchestrating Marketing

Duration 1:30 h

Educational Objectives:

- EO4.1 Know the core tasks of the functional area Marketing
- EO4.2 Know what contributions to make to core Marketing tasks
- EO4.3 Understand how a marketing department at a Software Producing Organization is organized
- EO4.4 Understand the key performance indicators of Marketing and how they align and conflict with your key performance indicators
- EO4.5 Understand the role of the marketing department in defining your market
- EO4.6 Know common sources of conflict with Marketing
- EO4.7 Apply understanding of common sources of conflict with Marketing to prevent potential issues

4.1 Introduction

Marketing is responsible for all aspects in preparation and support of the product sales activities of a company, including the creation of product awareness and communication of the positioning of the product in the market. Roughly speaking, their goal is crafting the most effective marketing and sales *funnel* as possible: attracting as many *strangers* as possible, turning them into *leads* that are interested in a product, converting them into paying *customers* and ultimately turning them into *promoters* of the product.

This product marketing goal closely aligns with one of three essential responsibilities of a software product manager as identified by Ebert (2007): *conquer markets and grow market share*. Because of this, effective collaboration with and orchestrating of the product marketing department is a core determiner for creating a winning product (Griffin and Hauser (1996)). Nevertheless, the core activities of the product management discipline drastically differ from those of product marketing. Product management defines the product to be built, while product marketing defines how to communicate the built product to the market.

Note that selling a product is not the most important part of marketing. Although sales is a closely related discipline, the goal of marketing should be to make sales unnecessary by developing the right product and communicating it to the market in the right way (Peter Drucker (1973), Kotler and Keller (2015)). Marketing achieves these goals by effectively communicating value propositions for specific personas. A value proposition is a promise of value to be delivered for a customer segment. Personas are fictional characters that represent an archetypical customer.

There are many different templates for defining a value proposition. Both Winer and Moore offer a simple and strong one:

For (target customer) who (need statement), the (product/brand name) is a (product category) that (key benefit statement/compelling reason to buy).

Unlike (primary competitor alternatives), (product/brand name) (primary differentiation statement)

Study ISPMA's syllabus "Excellence in Product Strategy" for more information on value propositions and personas.

4.2 The Marketing Organization

Marketing a software product needs to address the following three concerns (Kotler (2015)):

- *Strategic marketing*: defining an overarching strategic direction for marketing that is in accordance with the corporate strategy,
- *Product marketing*: translating the marketing strategy to the product level, and
- *Marketing communication*: executing the marketing strategies to create tangible deliverables, such as promotional materials or event booths with a strong focus on value communication.

Depending on the size and design of the organization, these concerns can be addressed by one single person, a team, or even multiple teams. The three concerns are highly interdependent. The marketing strategy influences the marketing goals that the product marketing team pursues. Similarly, the product marketing decisions impact marketing communication (MarCom). The success of the MarCom activities, then, affect the key performance indicators (KPI) of all three marketing disciplines. The MarCom KPIs inform the product marketing KPIs, which in turn inform the strategic marketing KPIs.

The chosen development methodology will impact the approach to orchestrating marketing. The typical marketing department will set plans, goals and budgets at the beginning of a new year or period and be unwilling to deviate from these plans in a significant manner. This lack of flexibility on the marketing side can create frustration in a high-pressure, quickly changing environment that is typical for a software development organization. Therefore, it is good practice to make agreements with the marketing department on the type of work and how much work they can do for a particular software product on an intermittent basis. For example, 10% of marketing resources could be made available to product management for announcing new product capabilities that were not yet committed at the time the annual marketing plan was finalized.

4.3 Typical areas of conflict

There can be some overlap between marketing KPIs and SPM KPIs. In this section, we further detail the overlap and how to achieve synergistic benefits. Despite these opportunities, marketing and product management frequently find themselves in a state of conflict. We highlight three common sources of conflicts and discuss ways to mitigate them.

Strategic conflict: brand versus product investment

A typical marketing department is inclined to invest in overall brand development because this will lead to a more beneficial impact for their KPIs. On the other hand, investment in product marketing suits the product manager's KPIs better. The decision of how to distribute marketing investments among brand development and marketing for specific products is a strategic decision that should be made at the board level. The product manager needs to raise this issue when necessary.

Product Conflict: Roadmap and Requirements

The roadmap is a heavily contested document that virtually all departments want to influence. The marketing department is no different. The number of interesting and potentially worthwhile ideas is typically much bigger than can be implemented, and a product manager needs to say no to most of these. At the same time, a product manager needs to be on the lookout for features that convey the 'allure of innovation'. Although these features may not drive sales, they demonstrate to the target customers that the vendor is an 'innovative player' or 'thought leader' that is worthy of their business.

Communication Conflict: Product Launch

The launch of a new product or product version requires careful orchestration of many different stakeholders within and outside of your organization. The marketing department is responsible for supplying all product promotion materials. However, asking them to drop everything and start working on a particular product when that product's next version is nearing technical completion will result in protest and conflict. It is best to make an organization-wide agreement on what is in a product promotion package and the lead time that's required for delivering such a package. Product promotion materials can take many different forms aside from the classical brochures and slide deck, such as:

- Press conference accompanied by a press release
- Media buys, both online and offline (Google ads, bus stop posters)
- Product logo
- Copy (i.e. text) for the website
- Public relations deals with strategic partners
- Inclusion in the organizational newsletter
- Social media presence

Approaches to addressing conflicts with marketing

To avoid and mitigate conflicts with marketing, a software product manager can:

- Ask the Board to define trade-off criteria for brand versus product marketing,
- Involve Marketing early and frequently in product strategy discussions,
- Agree what a product promotion package entails and the time required for completion, and
- Have a regular Jour Fixe meeting with Marketing to exchange information and make decisions in a timely way.

4.4 SPM's Focus Areas for Orchestrating Marketing

An SPM needs to focus on the following areas when orchestrating marketing:

- Positioning of product in marketing plan,
- Plan execution,
- Product launch,
- Value communication
- Channel and partner management,
- Selective participation in marketing events, and
- Trading off brand marketing with product marketing.

References: P. Drucker (1973); C. Ebert (2007); A. Griffin, J. R. Hauser (1996); P. Kotler, K. L. Keller, F. Ancarani, M. Costabile (2015); R. S. Winer (1999); G. A. Moore (2014); S. A. Fricker (2012); L. Weber (2017); H.-B. Kittlaus (2022)

EU5 Orchestrating Sales and Fulfillment

Duration: 1:30 h

Educational Objectives

- EO5.1 Know the key stages in a typical sales cycle, and their relationship to the customer buying cycle
- EO5.2 Understand how sales teams are organized
- EO5.3 Understand what motivates sales organizations and sales professionals and how this motivation may differ from SPM motivation
- EO5.4 Know the traditional points of intersection between software product management and sales and fulfillment activities and what each expects from the other
- EO5.5 Know the traditional sources of SPM/sales & fulfillment conflict and practices for minimizing and addressing them
- EO5.6 Understand the role of Fulfillment in delivering software products that aren't distributed on physical media

5.1 Introduction

The sales unit is responsible for all sales activities of a company, selling products directly to customers and channel partners. There are two approaches for a direct salesforce: **inside sales** and **outside sales**. Inside sales are made remotely with sales representatives interacting with customers via the phone or the internet. Outside sales typically involve interacting with customers face-to-face. As the internet grows in importance as a channel to software customers, and as more enterprise software is delivered as a service (SaaS), the role of inside sales is growing in some organizations.

Traditionally enterprise software vendors have targeted decision makers in IT departments and/or on the business side of customer organizations. More recently, there has been a trend to “product-led growth” (W. Bush 2019), also known as bottom-up sales. This means that potential users in enterprises become the primary target group. The product itself must be so convincing, intuitive and/or communicative that initial users are easily won over and become “sales agents” in their enterprises. This approach does not only require changes in Marketing and Sales, but in the product itself.

Outside sales are common in enterprise sales, e.g., B2B, where the sales process tends to be more complicated due to the complexity of the problem space and in customer decision-making. Because of the complexity of the collaboration required between outside sales and software product management, outside sales will be the focus of this syllabus.

Understanding the sales organization and related processes is important to software product managers because, as part of their orchestration responsibilities, they should attempt to influence the motivation of the sales organization in favor of their product, ensuring the sales force is willing to sell their product, and is enabled to do so.

5.2 Sales Motivation and Compensation

Compensation for sales professionals is different from that of other software organizational functions in that a significant proportion of their compensation is variable. Sales professionals are often compensated based on commission, which is a percentage of their sales revenue they generate. As part of the sales planning process, sales professionals can be given a “sales quota,” which is a monetary figure representing the minimum amount of revenue they are expected to generate from particular products or services. Setting quotas can be a powerful way to incent sales professionals to focus on specific products or markets.

Customers are sometimes given a discount on the purchase price of software products to encourage them to purchase it. Discounting is a powerful mechanism for sales professionals to convince customers to buy at all, or to accelerate buying decisions. Discounts, however, lower product-related revenue. We recommend strict governance rules for any pricing decisions (see ISPMA SPM EPS (2022)). In any case, Product Management must monitor the discounting practices and balance the need to close sales with defined business objectives to manage this trade-off.

5.3 Typical Areas of Conflict - Sales

Typical areas of conflict between SPM and sales include:

Getting Product Feedback from Sales

Contact with prospects, clients, and competitors make sales professionals a potentially valuable source of product-relevant information. However, the variable nature of their compensation and the inherent difficulty in the job may imply that there is insufficient incentive for members of the sales team to invest the necessary time to provide detailed feedback to the product organization. Product managers and their leadership must negotiate with sales leadership sufficient time for engagement, underscoring the unique value of information coming from the sales organization.

Sales Incentives

Through quotas and other incentives, sales professionals may not be motivated to give some products the attention the associated product managers expect. Product managers need to be aware of product-related incentives to understand whether their product is receiving the focus from Sales it needs to meet business objectives.

Sales Price and Discounting

To meet their sales targets for the current reporting period, salespeople may be inclined to grant significant discounts to close deals in their pipeline. However, excessive discounts lower a product's overall revenue and profitability. Product managers must monitor discounting policy and practice to ensure that overall business objectives aren't being unacceptably compromised to generate sales in the short term. Differing discounting policies between products can result in some products being heavily discounted to compensate for "discounting freezes" on others.

Short-term Customer Requirements vs. Longer-term Market Requirements

To secure sales, sales professionals may need to request features that a small number of customers (or even a single customer) requires. Product managers must take into account the opportunity costs of such investments, primarily related to diminished investment in features that are more broadly appealing. Features that are not widely used can generate significant development and maintenance costs over the life-cycle of the product, increasing complexity and eroding profitability. Helping sales to understand the business impact of investment alternatives and the cost impact of implementing features that aren't widely used can help product managers make convincing arguments.

Impact of SPM Customer Engagement on Sales

It is important or even critical for SPMs to have direct contact with customers for a variety of reasons, including gather feedback and validating product-related plans. This contact should be coordinated with sales to reduce the risk of endangering ongoing sales negotiations or the appearance that the different organizations within the software vendor are not aligned.

5.4 Approaches to Addressing Conflict - Sales

Product managers should seek to define a formal engagement model with Sales, including regular, timely face-to-face meetings to discuss strategic product-related topics

SPMs should:

- enlist the help of leadership to ensure the sales organization has the appropriate incentives in place to ensure to position and sell the product manager's product(s)
- regularly review discounting policy and actual discounts given, to understand the impact of discounting on their product's revenue
- ensure Sales understands the product strategy, in particular which customer needs the product is optimized for. This way Sales can better qualify customers and focus on those where the product is a good fit. This helps Sales to close deals faster, and reduces requests for "feature specials" that are needed only by one or very few customers
- actively keep the sales organization apprised of customer engagement activities, including updating CRM or similar systems of record, and personally informing account representatives (where applicable)

5.5 Fulfillment

Fulfillment means making the product available to the customer for use. Fulfillment can be under Sales' responsibility, or with a central fulfillment unit. Fulfillment can include stable and easy online orders, smooth and correct billing/payment, and offering downloads on the internet. Manufacturing is usually not an issue for pure software products or services except for shrink-wrapped software.

Making software available on a server for download is a common distribution method. The server infrastructure must be established and maintained, usually by an operations organization. SPMs must define expected availability and any authorization required for the software to be downloaded.

Representatives from a company's fulfillment function should provide fulfillment-related requirements to product management. These requirements may be related to supported pricing, licensing, and packaging models, required formats for digital assets, or physical properties of documentation and packaging. Professionals working with Fulfillment should also pass along input gathered from their interactions with channel partners.

5.6 Typical Areas of Conflict - Fulfillment

Fulfillment Planning and Scheduling

Fulfillment requires planning that can be impacted by changes to release schedule. Frequent changes to release schedules or poor communication can be a source of conflict between software product management and fulfillment as these changes may result in wasted efforts and additional cost.

Access Rights to Downloads

SPM must work with Fulfillment to ensure that people downloading the software product have appropriate authorization, including meeting licensing terms.

5.7 SPM's Focus Areas for Orchestrating Sales and Fulfillment

There are some areas that an SPM needs to focus on when orchestrating Sales and Fulfillment:

- Positioning of the product in the sales plan,
- Sales plan execution,
- Product-specific commitments to customers (features and measurements),
- Handling of customer requirements (short-term sales vs. longer-term product goals),
- Monitoring for deviations from standard terms and conditions,
- Monitoring for deviations from minimum price levels or price structure, or excessive discounting
- Selective participation in pre-sales meetings,
- Smooth and correct order, billing, and payment processes,
- Skills of sales representatives,
- In bigger companies: sales representatives dedicated to product family vs. cross-product, and
- Alignment of sales measurements with SPM's responsibilities (product vs. product group focus)

References: *ISPMA SPM EPS (2022); H.-B. Kittlaus (2022)*

EU6 Orchestrating Delivery Services and Support

Educational Objectives

- EO6.1 Know the core tasks of the functional areas delivery services and support
- EO6.2 Know what contributions to make to core delivery services and support tasks
- EO6.3 Understand how delivery services and support are organized
- EO6.4 Understand the role delivery services and support can play in positioning offerings with the customers they serve
- EO6.5 Recognize typical areas of conflict and formulate strategies to manage them

6.1 Introduction

The term “service” is overloaded in the software industry. In this syllabus, unless otherwise specified, we are referring to “human services,” providing customer value via people’s time and effort. More specifically, we are referring to services provided in the context of a software product. Because of their criticality to product success and the unique nature of the related orchestration activities, support services are covered separately later in this EU. Services offered by a company that are not relevant to its software products are beyond the scope of this syllabus.

Delivery Services mean all customer-specific services provided to customers to help them become productive with the initial deployment of the software product or whenever a new version is installed. This includes installation and tailoring services. Tailoring based on the product’s tailorability options is a customer-specific service that can entail a large project for configuration and customization.

Support refers to all product-related services provided to existing customers such as maintenance, training, operations, user help desk etc. Support provides technical support to customers, usually covered by maintenance or SaaS contracts. These services include education, installation, customization, operations, maintenance, technical support, and helpdesk covering technical and non-technical problems. Typically, these services are priced separately. Sometimes they are bundled with software product offerings.

Product managers need to have a broad and deep understanding of customer pains to identify opportunities to address customer challenges with services. Services can be critical for customer adoption and retention, particularly for products that address complex problems.

The vendor and partners can provide services. Even companies that would prefer to relegate service delivery to partners can be compelled by customers to provide them as well: some customers expect a complete solution and don’t want to establish an additional business relationship with a 3rd party service provider.

Although services can play an important role in providing maximum customer value, defining and managing services require knowledge and skills that are significantly different from those required to manage software products. While product managers should play a key role in identifying services opportunities relative to their product, i.e., service-related requirements, they should engage other professionals with the appropriate level of experience regarding detailed services definition, planning, pricing, and delivery.

6.2 Typical Areas of Conflict – Delivery Services and Support

Difficulty getting product feedback from Services

Because they often help customers deploy and use software products, Delivery Services personnel can be rich sources of feedback on the product. Unfortunately, they are often incented to spend as much

time as possible working on customer projects, leaving them and their leadership with little incentive to invest time in providing feedback or otherwise engaging with SPMs. SPMs should seek to develop relationships with members of the Delivery Services organization and should lobby for commitments to provide feedback to the product team. These commitments should include the definition of processes and guidelines for interfacing between the teams, budgeting a reasonable amount of time per year for the Delivery services organization to engage with the product team, and a suitable integration of the software tools used by the two teams.

Sharing the Product Roadmap/Positioning with Services

As previously noted, Services can spend a considerable amount of time with customers before and after the sale. They may therefore be able to position new features or offerings with customers. For this reason, product managers should invest time in sharing the roadmap and associated positioning with the Services organization.

Prioritizing Features for Markets vs. Individual Customers

The Delivery Services organization's involvement in customer projects may incent them to advocate for product features that have insufficient appeal to other customers. SPMs should be prepared to explain investment decisions to Service personnel, underscoring the necessity of meeting the needs of multiple stakeholders like customers.

Product Enablement for Delivery Services Personnel

Product managers should ensure that the Delivery Services personnel receive adequate orientation and training on products, especially in conjunction with new releases. Inadequate product enablement may result in suboptimal work on their part on customer projects or the perception that others, third-party services providers for example, have superior product knowledge.

Product Team Support for Delivery Services Organizations

The nature of the Service organization's work means that they may need support from the product team on solution design, bugs, etc. Product managers should attempt to provide the Delivery Services organization with the help it needs while minimizing the impact on the development team.

Ensuring Customer Satisfaction with Support Issue Resolution

Support professionals and organizations are often measured on how quickly they resolve customer issues. The KPI can incent them to mark support incidents as closed before the customer is satisfied with the resolution. Frequent engagement with the Support organization and guidelines or practices indicating when SPM should be made aware of important issues can help SPMs manage this risk.

Balancing Investment in Supportability Features with Other Priorities

Although making the product easy to support can be an important goal, such investments must be balanced with other investments that ensure appropriate market demand. All too often, investments in supportability are prioritized lower than other features customers request, resulting in suboptimal business performance due to support costs. SPMs should seek to assess investments in supportability based on business criteria.

Ensuring Support Professionals Have Adequate Knowledge of the Product

To adequately support the product, Support professionals must be trained on how the product functions, both from the end-user and the "back end" perspective. From a customer satisfaction perspective, it is in SPMs' best interest to ensure that the Support organization not only understands the technical aspects of the product – information they often get from engagement with Development – but understand the business purpose of the product. This gives them context that can be important in understanding the nature and magnitude of customers' support issues. The product team may need to supply product documentation aimed at Support professionals in addition to documentation intended for customers.

Timely Communication of Support Issues that Impact Customer Satisfaction

SPMs should work with the Support organization to define guidelines for when SPM should be made aware of issues that may have a significant impact on customer satisfaction. Issues that can eventually impact product business performance must be routinely brought to the attention of SPMs so the appropriate business decisions can be made. Since SPMs also often engage with customers, it is important that they not be caught unaware of critical issues.

6.3 Approaches to Address Conflict – Delivery Services and Support

SPMs should

- Create formal plans to ensure that Service personnel understands both the business/functional and technical aspects of the products, including positioning
- Define appropriate “handoffs” for customer queries regarding product roadmap, product strategy, and the value of the overall portfolio.
- Define reasonable practices to ensure that development capacity is not lowered by direct queries from Delivery Services and Support to Development
- Make themselves available to help Service personnel navigate the development organization as appropriate to ensure timely resolution of product-related issues encountered by the Service organization
- Make support professionals understand the important role they play in customer satisfaction, so they can better assess tradeoffs between this goal and support KPIs, such as closing open issues as quickly as possible
- Play an active role in upskilling Support with each new release, including the business motivation
- Consider investing some time working with or “shadowing” Support to better understand supportability requirements the daily challenges of the Support organization
- Actively seek supportability requirements from the Support organization and budget appropriately for those, for example in release planning
- Agree reasonably clear criteria with the Support organization when to contact the SPM regarding relevant support issues

6.4 SPM’s Focus Areas for Orchestrating Delivery Services and Support

There are some areas that an SPM needs to focus on when orchestrating Delivery Services and Support:

- Consider and manage product-related services and documentation as part of the offering
- Service execution
- Skills of Service specialists
- Frequent analysis of incoming service calls: these are often good indicators for problems with quality, usability, and functional coverage
- Resource management: avoid bottlenecks that impact Product Development, Sales, and customer satisfaction

References: H.-B. Kittlaus (2022)

EU7 Complex Scenarios Involving Multiple Functions

Duration: 1:30 h

Educational Objectives:

- EO7.1 Understand that software product managers may encounter situations where their work impacts multiple other functions simultaneously
- EO7.2 Apply the ISPMA® reference framework to understand which activities of other functions are affected by the current situation and to identify the required stakeholders
- EO7.3 Apply the orchestration techniques from EU1 and EU2 and skills in orchestrating other functions from EU3 to EU6 to successfully collaborate with other functions in these complex situations

As part of their everyday work, software product manager will encounter situations where their work impacts multiple other functions.

This multi-impact situation often occurs with product strategy decisions. For example, a change in pricing may affect the pricing metric. The changed metric may require Development to implement changes in license reporting or enforcement, Support to update support pricing, Marketing to update the price list and license key generation and to communicate the change to customers, and Sales to be re-trained.

A multi-impact situation can also be related to operational tasks. For example, resolving a customer problem that has been escalated may require the product manager to change their roadmap to provide a good longer-term solution, and to coordinate activities between Support, Development, and Sales (account management for the customer) to find an acceptable workaround for the short term.

The ISPMA® reference framework (see Fig. 1) can be used to understand which activities of other functions – represented by cells in the orchestration columns – are affected by the decision or problem at hand. The knowledge of affected cells can help identify the stakeholders that need to be involved.

Once the right stakeholders have been identified, the software product manager needs to use their understanding of the other functions (from EU3 to EU6) to map out a suitable approach for addressing the situation. For example, this may include establishing role clarity and defining a process to use to address the situation at hand. Then, the product manager can exercise their leadership skills to work through the agreed process with the stakeholders: remembering the mindsets of influential product managers, leveraging their sources of power, and using techniques for conflict management and negotiation.

References: H.-B. Kittlaus (2022)

Bibliography

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

Bass, L., Weber, I., Zhu, L., (2015): DevOps: A Software Architect's Perspective. Addison-Wesley, Upper Saddle River

Bech, H.P. (2015): Building Successful Partner Channels. TBK Publishing

Bush, W. (2019): Product-Led Growth: How to Build a Product That Sells Itself. Product-Led Institute

Cagan, M. (2008): Inspired - How to Create Products Customers Love. SVPG Press

Drucker, P. (1973): Management: Tasks, Responsibilities, Practices. Harper and Row, pp. 64–65.

Ebert, C. (2007): The Impacts of Software Product Management, Journal of Systems and Software, vol. 80, no. 6, pp. 850-861

Erich, F., Amrit, C., & Daneva, M. (2017): A Qualitative Study of DevOps Usage in Practice. Journal of Software Evolution Process

Fisher, R., Ury, W.L., Patton, B. (2012): Getting to Yes: Negotiating Agreement Without Giving In, upd. rev. ed.. Penguin

Fricker, S.A. (2012): Software Product Management, in: Maedche, A., Botzenhardt, A., Neer, L. (eds.): Software for People - Fundamentals, Trends and Best Practices, Springer, Heidelberg, pp. 53-81

Fricker, S.A., Grünbacher, P. (2008): Negotiation Constellations – Method Selection Framework for Requirements Negotiation. International Working Conference on Requirements Engineering: Foundation for Software Quality (RefsQ 2008), Montpellier, France

Gothelf, J., Seiden, J. (2013): Lean UX - Applying Lean Principles to Improve User Experience. O'Reilly

Griffin, A., Hauser, J. R. (1996): Integrating R&D and Marketing: A Review and Analysis of the Literature, Journal of Product Innovation Management, vol. 13, no. 3, pp. 191-215

Haines, S. (2021): The Product Manager's Desk Reference, 3rd ed.. McGraw-Hill

Hall, K. (2013): Making the Matrix Work - How Matrix Managers Engage People and Cut Through Complexity. Nicholas Brealey Publishing

Humble, J., Farley, D. (2010): Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation. Addison-Wesley

Humble, J., Molesky, J. (2011): Why Enterprises Must Adopt DevOps to Enable Continuous Delivery, Cutter IT Journal, vol. 24, no. 8

ISPMA SPM F (2022): ISPMA® SPM – The Foundation Syllabus, V.2.1, 2022

ISPMA SPM EPS (2022): ISPMA® SPM Excellence in Product Strategy Syllabus, V.2.1, 2022

Jabbari, R., bin Ali, N., Petersen, K., & Tanveer, B. (2016): What is DevOps? A Systematic Mapping Study on Definitions and Practices. Proceedings of the Scientific Workshop Proceeding of XP'2016

Kittlaus, H.-B. (2022): Software Product Management, The ISPMA-Compliant Study Guide and Handbook, 2nd Edition. Springer.

Knaster, R., & Leffingwell, D. (2020): SFA® 5.0 Distilled: Achieving Business Agility With the Scaled Agile Framework®. Addison-Wesley

Kotler, P., Keller, K. L., Ancarani, F., & Costabile, M. (2015): Marketing Management 15/e. Pearson

- Leffingwell, D. (2011): Agile Software Requirements, Addison-Wesley*
- Lewrick, M., Link, P., Leifer, L. (2020): The Design Thinking Toolbox: A Guide to Mastering the Most Popular and Valuable Innovation Methods. Wiley*
- Moore, G.A. (2014): Crossing the Chasm, Harper, New York*
- Pichler, R. (2020): How to Lead in Product Management - Practices to Align Stakeholders, Guide Development Teams, and Create Value Together. Pichler Consulting*
- Rahim, M. A. (2015): Managing Conflict in Organizations, 4th ed., Transaction Publishers*
- Sandy, K (2020): The Influential Product Manager. Berrett-Koehler Publishers*
- Schwaber, K., Sutherland, J. (2020): The Scrum Guide - The Definitive Guide to Scrum: The Rules of the Game. <https://www.scrumguides.org>*
- Torres, T. (2021): Continuous Discovery Habits, Product Talk LLC*
- Voss, C. (2016): Never Split The Difference, Penguin Random House*
- Weber, L. (2017): The Product Marketing Manager, Amazon*
- Weiss, J., Hughes, J. (2005): Want Collaboration? Accept - and Actively Manage – Conflict, Harvard Business Review, vol. 83, no. 3, pp. 93-101*
- Winer, R. S. (1999): Marketing Management, Prentice Hall, p. 64*
- Zartman, I. W., Rubin, J. Z. (2000): The Study of Power and the Practice of Negotiation, in: Zartman, I. W., Rubin, J. Z. (eds.): Power and Negotiation, University of Michigan Press, 2000, pp. 3-28*

Internet Links

- IL1: <http://martinfowler.com/bliki/ContinuousDelivery.html>
- IL2: <http://scaledagileframework.com/>
- IL3: The Agile Admin: What is DevOps? <http://theagileadmin.com/what-is-devops/>; 2011 (Ernest Mueller, James Wickett, Krthik Gaekwad, Peco Karayanev)