



ISPMA

INTERNATIONAL
SOFTWARE PRODUCT MANAGEMENT
ASSOCIATION

Software Product Management for Startups

For software startups as well as new software product businesses inside mature business enterprises

Student Edition

V.1.1

Copyright © ISPMA® 2023

This SPM for Startups syllabus is an adaptation of the original syllabus “SPM – The Foundation” V.2.1 which was written by the following members of the International Software Product Management Association (ISPMA®): Gerald Heller (editor), Jonas Als, Magnus Billgren, Erik Bjernulf, Sjaak Brinkkemper, Christof Ebert, Samuel Fricker, Tony Gorschek, Rainer Grau, Marc Hilber, Barbara Hoisl, Slinger Jansen, Mahvish Khurum, Hans-Bernd Kittlaus, Daniel Lucas-Hirtz, Andrey Maglyas, Sacha Reis, Niklas Rosvall, Kevin Vlaanderen and Inge van de Weerd.

The SPM for Startups syllabus has been written by Haragopal Mangipudi (editor), Samuel Fricker, Barbara Hoisl, Lucas Weber, Frédéric Pattyn and Hans-Bernd Kittlaus.

We thank all honorary authors and contributors.

Copyright © 2023 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 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 “SPM for Startups” syllabus covers the full spectrum of elements for software product management as relevant and applicable to startups as well as New Product businesses in mature companies, that are well-supported by literature and industrial practice. The syllabus corresponds to a two day (12 hours of Instruction) industry course.

The syllabus addresses the needs of people involved in software product management in the startups as well as those building new software products in mature businesses, including those that the product manager interfaces with, e.g. founders, investors, incubators, accelerators, general management, marketing and sales, research and development, production, service and support, and controlling.

This syllabus is explicitly focused on software product management – hence, concepts and practices of entrepreneurship are out of the scope for this syllabus. This framework is agnostic to the nature of startup as well as the leadership style of the founder. Therefore, whether the startup is a social enterprise or for profit, whether it is funded or bootstrapped, whether it is a venture or a lifestyle, whether it is a new product in a large company or a new product in a new business, our syllabus and framework address the focal points of software product management function in those contexts in establishing the business model by addressing the mutual fit between the product and market.

The syllabus is the basis for an examination to certify that the examinee has achieved the degree of knowledge described in this syllabus.

Purpose and structure of the Syllabus:

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

- Explicit educational objectives for each chapter
- 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 four 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 and subchapter numbering. Educational objectives that apply to the entire EU use a „0“ as second digit.

The educational objectives are expressed in terms of two cognitive levels of knowledge, expressed with the verbs “know” for level 1 (L1) and “understand” for level 2 (L2). 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:

This SPM for Startups syllabus covers knowledge applicable for any kind of software systems and organizational contexts related to building new software products business either in startups or in mature business enterprises. 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.

This 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.

For the rest of this syllabus document, unless otherwise specified, the use of “startup” includes software startups as well as new software product businesses inside mature business enterprises. A detailed description of a startup and the evolution of a product across the stages of a startup are given in section 1.1 and fig. 1 in this syllabus.

Training Courses:

The syllabus corresponds to a two-day in-person or three-day digital 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 “ISPMA® Certified Software Product Manager – **Startups** “ certification examination.

All chapters are relevant for the exam. The exam takes the following form:

- Demonstration of knowledge and its application 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 startup ecosystem and building software products anew in a startup or in a corporate venture.

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. This syllabus assumes basic awareness of the foundations and terminology of entrepreneurship and its practices in various contexts.

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 and Foundations	
	1.1 Software Product Management Essentials for Startups	0:30 h
	1.2 Startup Software Product Management Framework	0:30 h
EU2	Product Strategy	
	2.1 Product Strategy Essentials	0:30 h
	2.2 Product-Market Fit	0:30 h
	2.3 Positioning and Product Definition	0:30 h
	2.4 Delivery Model and Service Strategy	0:15 h
	2.5 Sourcing	0:15 h
	2.6 Pricing	0:45 h
	2.7 Financial Management	0:45 h
	2.8 Ecosystem Management	0:30 h
	2.9 Legal and IPR Management	0:30 h
	2.10 Performance and Risk Management	0:15 h
EU3	Product Planning	
	3.1 Product Planning Approaches	0:30 h
	3.2 Customer Insight	0:30 h
	3.3 Product Requirements Engineering	1:00 h
	3.4 Release Planning	0:30 h
	3.5 Roadmapping	0:30 h
	3.6 Product Life Cycle Management	0:30 h
EU4	Strategic Management & Functional Areas	
	4.1 Corporate Strategy and Organizational Management Essentials	0:15 h
	4.2 Competitive Strategy	0:15 h
	4.3 Funding	0:15 h
	4.4 Compliance Management	0:15 h
	4.5 Market Analysis	0:15 h
	4.6 Initiation and Management of Functional Areas	
	4.6.1 Development	0:30 h
	4.6.2 Marketing	0:30 h
	4.6.3 Sales and Fulfillment	0:15 h
	4.6.4 Delivery Services and Support	0:15 h

Bibliography

EU1 Introduction and Foundations

Duration: 1:00 h

Educational Objectives:

EO1.1.1 Know the scope and effects of software product management in startups

EO1.1.2 Know the function and objectives of software product management in startups

EO1.2.1 Know the SPM Framework for startups

EO1.2.2 Understand the multidisciplinary nature of SPM

1.1 Software Product Management Essentials for Startups

Duration: 0:30 h

Product management has become an established discipline in many industries since Procter & Gamble introduced it in 1931 as brand management. During the last decades, many software product companies – such as Microsoft, IBM, and SAP – 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. Software product management 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 releases in terms of schedule predictability and time-to-market.

SPM has become critical in the early stages of startups to ensure the necessary focus on business requirements, strategy, ecosystem positioning and customer experience during the early phases of the company and product. The younger the startup, the greater the possibility that “persevere or pivot” decisions will be made to the product, in which SPM is a key function. Wherever dedicated SPM is not hired, one of the founders may hold the product management function or founder(s) or early members of the team could share the product management responsibilities amongst themselves.

A startup is not a smaller version of a business enterprise. To give an analogy, a tadpole and a caterpillar are not the smaller versions of a frog and a butterfly respectively. A startup is a temporary organization designed to search for a repeatable and scalable business model. A startup is a human institution designed to deliver a new product or service under conditions of extreme uncertainty.

Hence, based on the given vision and strategy the purpose of software product management function in a startup is to build and update business model and strategy by devising, conducting and executing experiments and using the learnings from each of such iterations.

This is the fundamental difference in the purpose of the software product management function in startups compared to the software product management function in a mature business.

Therefore, all references to the role of “Software Product Manager “ in this syllabus document should be read as applicable to Product Management function in a startup organization or for a new product in a mature business.

In the context of Software Product Management, startups include all the three variants below who embark on the journey of building a new product to address a pressing problem of a market segment:

- Bootstrapped startups
- Funded Startups
- Innovation Labs / New Product Businesses inside mature companies (This syllabus is relevant only for the stand-alone new product businesses inside mature companies).

We differentiate startups in three specific stages:

1. Early stage startups
2. Growth stage startups
3. Scale-ups

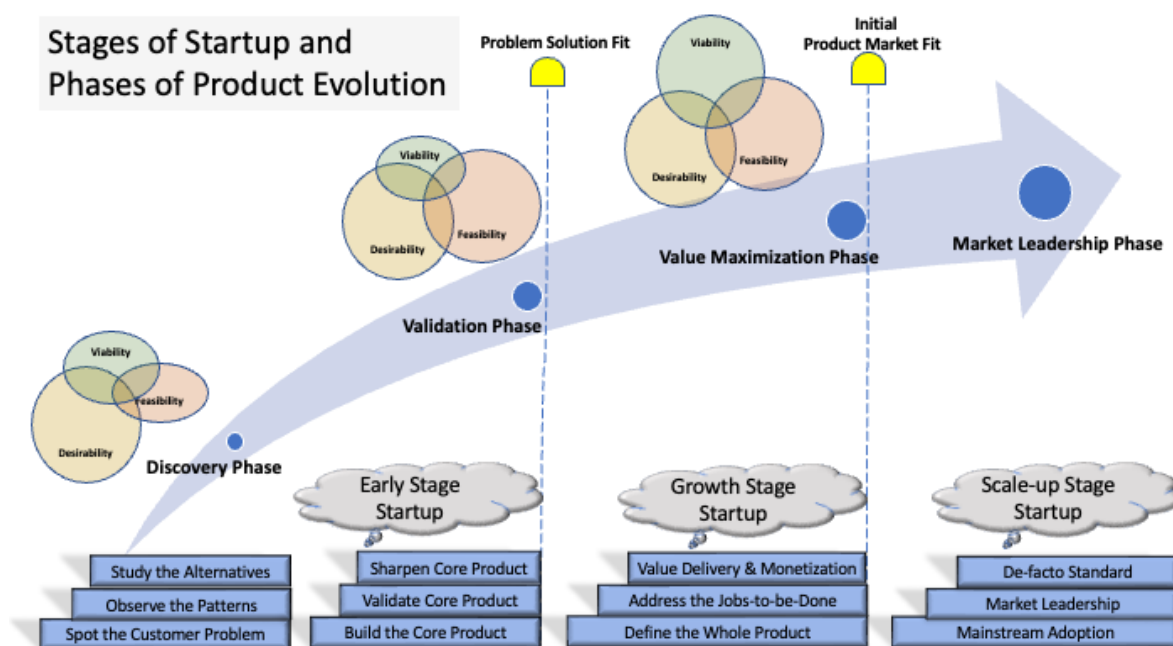


Fig. 1 Stages of a startup
 (Kittlaus, H.-B., & Mangipudi, H. (2023))

The Early Stage is focused on finding the right Problem-Solution Fit which aims at deriving a minimum feature set in which to launch a minimum viable product, an MVP, which is to be the subject of learning and iterated upon toward solving the customer's confirmed problem.

Literature: *Maurya, A. (2012)*

The Growth Stage is focused on finding and optimizing the product-market fit using an iterative approach. The steps of an iteration are as follows: hypothesis – MVP – test – conclusion. Eric Ries

defined MVP as “That version of a new product which allows a team to collect the maximum amount of validated learning about customers with the least effort.” Since the aforementioned definition of an MVP lacks a customer value perspective, we prefer the following definition: „The minimum feature set of a new product that is derived through a learning phase and that some customers are willing to pay for in the first release.”

Literature : *Blank, S. and Dorf, B.(2012 p. xvii ff)*

The learning process in these early stages can sometimes result in drastic decisions, so-called pivots that change the course of the product and startup. Pivots can be of three types:

- **Product Pivot:** When the startup learns that one or some parts of the product are perceived as way more valuable than the rest of the product and the customers care more or are willing to pay more for those.
- **Customer Pivot:** When startup learns that there is a new segment of customers willing to pay more for your product than your present customers.
- **Problem Pivot:** While talking to customers and doing the research, startup discovers that the customers have much bigger problem than the startup is trying to solve.

Stages 1 and 2 are addressed by this syllabus. In stage 3, the startup has a significantly increased maturity so that the syllabus “SPM – The Foundation” is applicable.

Literature: *Ebert, C. (2007); Fricker, S. (2012); Ries, E. (2011, p. 27); Moore, G. (2014); Riani, A. (2019); Areitio, A. (2019), Kittlaus, H.-B., & Mangipudi, H. (2023).*

A product is a combination of goods and services, which a supplier/development organization combines in support of its interests and where defined rights are transferred to a customer. A software product is one whose primary component is software. It contrasts with physical products in terms of high complexity, negligible manufacturing cost, great flexibility, and a high rate of change. In most cases, the interests of an organization are commercial, but there are exceptions, as is sometimes seen in open source-based products or in the public sector. Software products are typically produced for multiple (customer) organizations with many users.

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 services, can also be products from all industries like financial, insurance, gaming, social software, or human 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: *Kittlaus, H.-B., & Mangipudi, H. (2023)*

A platform in an ecosystem is a technical product that the platform owners as well as third parties use as a foundation for conducting their own respective businesses. This may include technological

collaboration and integration, as well as commercial interaction and financial transactions. There are two non-disjoint types of platforms:

- Innovation platform: Technological foundation upon which the owner and other firms develop complementary innovations.
- Transaction platform: Intermediary or online marketplace that makes it possible for people and organizations to share information or to buy, sell, or access a variety of goods and services.

A software product manager is responsible for managing software with the objective to achieve sustainable success over the life cycle of a software product. 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 have to manage a broad set of product-related activities as shown in the ISPMA® SPM Framework (Figure 2). They have to act proactively and be the responsible and engaged driver of their products. 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 objectives. This is especially true within the startup culture in which the SPM may be required to wear many “hats” and therefore, take on more responsibilities than would be required in larger and more established organizations.

Literature: *Cusumano, M. (2004); Cusumano, M., et al. (2019); Ebert, C. (2007); Fricker, S. (2012); Kittlaus, H.-B., & Mangipudi, H. (2023); Parker, G.G., et al. (2016); Kittlaus, H.-B. (2020b); Ries, E. (2011).*

1.2 Software Product Management Framework for Startups

Duration: 0:30 h

The SPM framework for startups provides a holistic view on the activities 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 “Customer Relationship Management” (CRM) that contain both longer-term and shorter-term aspects.



Fig. 2 SPM Framework for Startups V.1.0

Some elements of the SPM Framework are relevant in the early stage (E), some only become relevant when a startup enters the growth stage (G). Some topics need special attention in the Early and Growth Stages of a startup. They are added as elements to the framework. Competitive Strategy and Customer Experience Design apply from the early stage of the startup while Funding and Product-Market Fit from growth stage onwards.

This SPM Startup Syllabus is structured in correspondence with the SPM Framework for Startups with only a few deviations due to pedagogical reasons. It starts with Product Strategy and Product Planning, then Strategic Management followed by other Functional Areas.

The SPM framework for Startups captures the three-dimensional focus of the startup business namely, product readiness, market readiness and organizational readiness.

In the early and growth stages of a startup, not all corporate functions may be fully staffed. The software product manager may play a part in those roles until they become fully functional roles - thereby the software product manager's role shifts from participation to orchestration.

Literature: Ebert, C. (2007); Ebert, C., & Brinkkemper, S. (2014 pp. 10-18); Fricker, S. (2012); Kittlaus, H.-B. (2022, p. 37 ff); van de Weerd I., et al. (2006), Kittlaus, H.-B., & Mangipudi, H. (2023).

EU2 Product Strategy

Duration: 4:45 h

Educational Objectives:

- EO2.1.1 Understand the role of a product strategy, its elements and their interdependencies
- EO2.2.1 Understand the elements of the product definition
- EO2.2.2 Understand product positioning and its elements
- EO2.3.1 Understand the software delivery models and their consequences
- EO2.3.2 Understand the importance of integrating the service strategy into the product strategy
- EO2.4.1 Know the nuances of Product-Market Fit
- EO2.5.1 Know the different types of sourcing
- EO2.6.1 Know the Strategic Pricing Pyramid
- EO2.6.2 Know the importance of customer value and how it can impact pricing
- EO2.7.1 Know elements relevant for financial evaluation and investment decisions
- EO2.7.2 Understand the business model concept
- EO2.8.1 Know the role, significance, and elements of a software ecosystem
- EO2.9.1 Understand the importance of protecting intellectual property
- EO2.9.2 Know intellectual property protection mechanisms
- EO2.10.1 Understand startup metrics for the business performance of the product

2.1 Product Strategy Essentials

Duration: 0:30 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 and life cycle stage.

The product strategy describes how the product is supposed to evolve over this strategic time frame. It should address the following items:

- Positioning and Product Definition including Product-Market Fit
- Delivery model and Service Strategy
- Sourcing
- Pricing
- Financial Management
- Ecosystem Management
- Legal and IPR Management
- Performance and Risk Management

A product strategy document is normally very fluid for startups and more often represented in schematic representations for easy comprehension and upkeep. All of these items are highly interdependent. In a startup situation, the development of a product strategy is a highly iterative process that aims at finding and optimizing the product-market fit (see sections 1.1 and 2.2). Through this iterative process, development and implementation of a product strategy are intertwined.

To engage and align all stakeholders, a strong product vision driven by the larger purpose of the startup / enterprise 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.

Literature: *Kittlaus, H.-B., & Mangipudi, H. (2023); McGrath, M. (2000); Osterwalder, A., & Pigneur, Y. (2010); Popp, K.M. (2015); Pichler, R. (2016, Part I)*

2.2 Product-Market Fit

Duration: 0:30 h

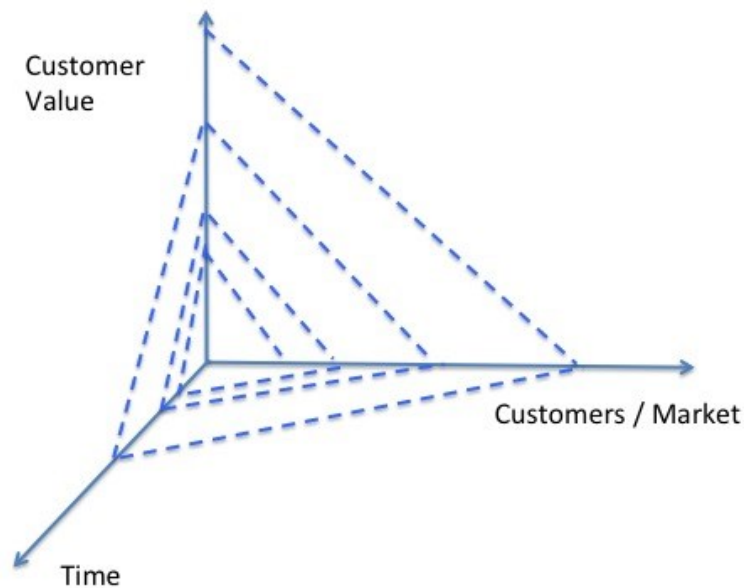
Finding Product-Market Fit is an iterative process of identifying a compelling value proposition based on value hypotheses that are tested (see Fig. 3).

A value hypothesis is an attempt to articulate the key assumption that underlies why a customer is likely to use your product. Identifying a compelling value hypothesis is indeed the finding of the product-market fit. A value hypothesis addresses both the features and business model required to entice a customer to buy your product.

A technique that is frequently used for this iterative process is continuous discovery.

A growth hypothesis represents your best thinking about how you can scale the number of customers attracted to your product.

Literature : *Radcliff, A. (2013); Andreesen, M. (2007); Kittlaus, H.-B., & Mangipudi, H. (2023)*



The Virtuous Spiral of Product-Market Fit

Product-Market Fit is not an event but a continuous journey of market relevance

© gunaka

Fig. 3 Virtuous Spiral of Product-Market Fit

2.3 Positioning and Product Definition

Duration: 0:30 h

Product positioning includes

- Value Proposition: value definition from a customer perspective for the target market segments.
- Focus with regard to the target market and segments
- Channel options
- Partnerships and alliances

The product definition needs to define:

- Functional scope
- Quality scope
- Intended use and users
- Customer experience (CX) and User experience (UX) design scope
- Offering architecture (see section 4.6.1)

- Business architecture (for application software) (see section 4.6.1) Value Proposition, which is also called customer or product value, is the benefit that a customer gets by using a product to satisfy her/his needs. In value definition exercise SPM should distinguish the following:
 - Absolute vs Relative Value
 - Real vs Perceived Value
 - Ideal, Diminishing and Declining Value
 - Value of a habit

Value of habit is of particular importance for startups that attempt to disrupt prevailing consumption or usage behaviors. Startup may need a stronger value proposition to address the consumption or usage inertia.

The offering architecture defines separately priced components of the product offering, and tailorability options in line with the tailorability strategy (see section 2.3). The business architecture is relevant for application software, and is domain-specific, i.e. covers logical data model, process model, business object model etc.

In a startup situation, initially there may not be a position with the title “business architect”, but one or more people with appropriate skills, e.g. product management, need to take care of architectural decisions early on.

When working on the product definition, the product manager has to take compliance aspects into account (see EU4), e.g. in the areas of sustainability and ethics.

A startup with its limited resources needs to collaborate with a diverse set of product and service partners in the ecosystem to complete the “whole product” which provides a solution to the needs and aspirations of the target customers. Thus the product definition as well as the value proposition should consider the “whole product “ (see 2.8 Ecosystem Management).

For defining the target market and identifying potential segments, the following elements need to be addressed:

- Scope, size, competitive products and their market shares, including international markets and their opportunities. (See “Market Analysis” in 4.5)
- Segmentation: should a standard market segmentation (e.g. from industry analysts) be used or is a custom segmentation required?

In a startup, product positioning typically defines the entire company positioning as well. Product positioning and definition are done iteratively to achieve the right product – market fit.

Literature: *Bech, H.P. (2015); Dunford, A. (2019); Kittlaus, H.-B., & Mangipudi, H. (2023); McGrath, M. (2000); Almquist, E., et.al. (2016, pp. 46-53); Osterwalder, A., & Pigneur, Y. (2014)*

2.4 Delivery Model and Service Strategy

Duration: 0:15 h

Based on the product definition, the delivery model needs to address the following items:

- Licensed product vs. service offering (e.g. Software-as-a-Service (SaaS))
- Mode of delivery (online access, online download, combination with services, etc.)

The service strategy needs to define the services that are part of the total offering, and who are supposed to provide these services.

Literature: *Kittlaus, H.-B. (2019); Kittlaus, H.-B., & Mangipudi, H. (2023); McGrath, M. (2000); Kowalkowski, C., & Ulaga, W. (2017, Ch.1,2 & 4)*

2.5 Sourcing

Duration: 0:15 h

Inorganic expansion of product and/or market footprint can be found in high growth startups and scale-ups. The Product Management function plays an important role both in the pre and post acquisition process for the strategic acquisitions – predominantly in value amplification. Sourcing needs to address make or buy decisions for software components.

There are a number of motivations for not developing a piece of software internally, like:

- Focus on time to market
- High quality and low cost of an externally available software product
- Restrictions with regard to internal capacity, technology, and skills

The dependency on an external software provider resulting from a buy decision needs to be managed thoroughly from a business perspective.

Literature: *Borg, M. et al. (2019); Kittlaus, H.-B., & Mangipudi, H. (2023); McGrath, M. (2000)*

2.6 Pricing

Duration: 0:45 h

With regard to Pricing the following items need to be taken into account:

- The importance of price with regard to business success and customer value
- Market- and value-based pricing
- Problems of cost-based pricing for software offerings
- Strategic Pricing Pyramid (price strategy, policy, level) (see figure 4)
- Typical pricing models for software including freemium

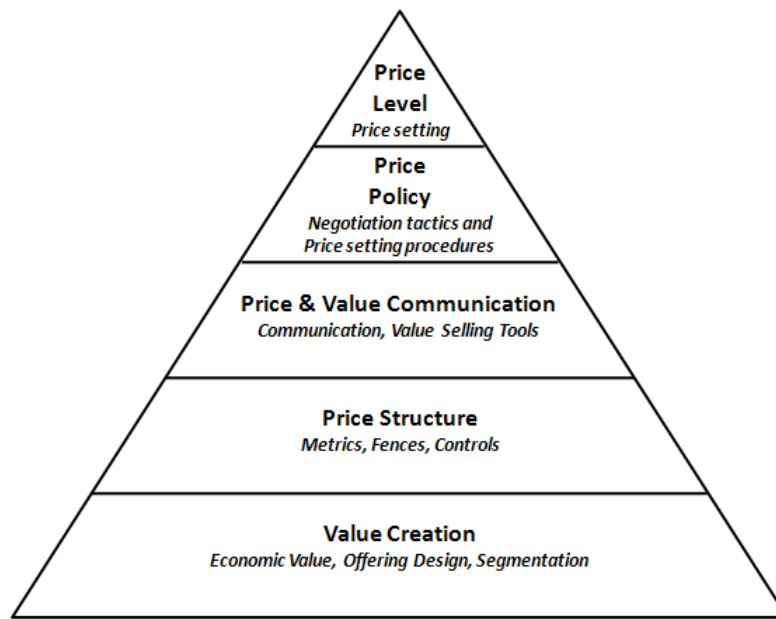


Fig. 4 Strategic Pricing Pyramid (© Nagle/Hogan 2005)

Startups have a few additional challenges as well as opportunities in Product Pricing strategies as compared to mature product businesses:

- Price optimization for Demand vs Revenue vs Margins
- Price discovery where there are no comparable competitive products or alternatives.
- Value mapping and articulation for both setting and getting the price

Literature: Kittlaus, H.-B., & Mangipudi, H. (2023); Kittlaus, H.-B., & Clough, P. (2009, p. 117 ff); Nagle T.T., et al. (2014); Nagle T.T., & Hogan J.E. (2005); Sodhi M.N., & Sodhi, N.N. (2007)

2.7 Financial Management

Duration: 0:45 h

The primary objective of software product management is to achieve sustainable economic success over the life cycle of the product. 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 to plan and track financial aspects both from a short- and long-term perspective. This is called financial management and is tightly linked to pricing.

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. For software, the most relevant business models are the software product vendor, either with license products or with SaaS products, and the professional service provider.

For startups, Lean Canvas, an adaptation of the Business Model Canvas is used to build and articulate the proposed business model of the startup around the customer problem that the startup intends to address and the proposed solution. The canvas defines and visualizes a business model by integrating positioning and product definition, product – market fit, delivery model and service strategy, pricing, financial model and ecosystem which are product strategy cells of the ISPMA® Framework. Unlike an established business a startup has to create a niche for itself by taking on the incumbents as well as scores of alternatives the target customers may have. For this, startups need a competitive differentiation or Unique Value Proposition. Secondly, the startup should sustain such differentiation with sufficient moat or unfair advantages so that the larger incumbents / competition cannot catch-up easily. Unique Value Proposition and unfair advantage in the canvas are mapped to Competitive Strategy in the ISPMA® Framework.

The financial models for venture-capital-oriented startups differ from steady-state mature products. Founders as well as Product Management stay highly focused on not just the product value, but even more on the valuations. Thus the process of prioritization of market segments as well as the corresponding product choices will add the third dimension of time horizon to the cost-value mapping.

Literature: *Osterwalder, A., & Pigneur, Y. (2010); Chen, S., et al. (2013); Kittlaus, H.-B., & Mangipudi, H. (2023); Haines, S. (2014, Ch.6)*

The Business Plan builds on:

- Validation of the Problem-Solution Fit
- Expanding the addressable customer segments and achieve Product-Market Fit
- Expanding the reach through sales channels
- Drive the customer acquisition and user adoption

Literature: *Kittlaus, H.-B., & Mangipudi, H. (2023); Schmidt, M. (2002); Sheen, R., Gallo, A. (2015); Haines, S. (2014); Osterwalder A., & Pigneur Y. (2010)*

2.8 Ecosystem Management

Duration: 0:30 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 while maintaining beneficial relationships. 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: controlling the ecosystem from the strategic position of platform owner, but leaving space for other actors
- Dominator: controlling the ecosystem from the strategic position of platform owner, striving for complete control

- Niche player: not the platform owner, but benefitting from the ecosystem through specialization and avoiding conflict with keystones or dominators

Startups most often do not have the assets and/or time to build the “whole product” that meets the complete needs of the target market. Secondly, startups need the “giant shoulders” of ecosystem partners to access the market segments wider and deeper in the shortest possible time. As Geoffrey Moore describes the “Bowling Alley” phase of the technology adoption life cycle requires a host of ecosystem products and services offerings to complement the “core product” or the Minimum Viable Product. Startup SPM plays an active role in defining the contours of the whole product around the core product and also in shortlisting the partner offerings to fit the white spaces.

Literature : *Kittlaus, H.-B., & Mangipudi, H. (2023); Moore, G. (2014); Ries, E. (2011)*

2.9 Legal and IPR Management

Duration: 0:30 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 responsible for the sustainable success of the product. First, there are contractual issues between the software vendor and the customer. Then, there is the protection of intellectual property. Third, there are specific risks such as product liability, data protection (especially for SaaS, in Europe in particular the General Data Protection Regulation (GDPR)), open source license conditions, etc.

The contract, by which software is “acquired”, may be individually negotiated or, particularly in the mass-market, based on so-called “terms and conditions,” which are subject to regulations, in particular if the customer is a consumer. Be it in a contract or in standard terms and conditions, the required legal terms, include:

- Scope of the license or service
- Guarantee and Warranty / SLA
- Type of charges
- Liability
- Maintenance provisions
- Miscellaneous legal provisions

Literature: *Kittlaus, H.-B., & Mangipudi, H. (2023)*

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. Software is primarily protected by copyright. The copyright protects the source code and object code in its given form. Copyright does not protect the underlying ideas, functions and algorithms and a copyright infringement thus presupposes an unchanged use of the software code (or parts

thereof). Product material such as manuals, brochures and product presentations are usually also protected. There are four more fundamental legal constructs, which may play a role for the protection of intellectual property:

- Trademark:** Protection for the names of brands, i.e. trademarks 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 method for software protection. 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 generally only be obtained for software which is integrated into a technical solution to solve a problem.

Literature: *Kittlaus, H.-B., & Mangipudi, H. (2023); Klemens, B. (2006)*

2.10 Performance and Risk Management

Duration: 0:15 h

Performance management involves continuous tracking and analysis of selected relevant measures and taking timely action if needed.

Due to the continuous state of evolution, startups need a different set of performance metrics than mature product businesses. To facilitate the iterative validation learning, the performance metrics in startups should be accessible, auditable and actionable.

Acquisition, Activation, Retention, Revenue and Referrals are some of the common product metrics that are used in startups to measure the product-market fit.

Risk management does usually not get much attention in the early stages of a startup.

Literature: *Kittlaus, H.-B., & Mangipudi, H. (2023); McGrath, M. (2000); Pritchard, C.L. (2015); Schmidt, M. (2002);*

EU3 Product Planning

Duration: 3:30 h

Educational Objectives:

- EO3.1.1 Understand the three different approaches to product planning
- EO3.1.2 Understand the different pivot types
- EO3.2.1 Know typical activities how product managers can gather customer data
- EO3.3.1 Understand the specifics of product RE
- EO3.3.2 Understand the importance of understanding the customers
- EO3.4.1 Know the purpose of release planning
- EO3.5.1 Know the concept, importance and elements of a roadmap
- EO3.5.2 Understand the various ways roadmaps are used and communicated
- EO3.6.1 Know the phases of the product life cycle

3.1 Product Planning Approaches

Duration: 0:30 h

Product planning is the combination of processes used for converting the product strategy and external insights into an executable plan that is the basis of the product team's work.

For a long time, the standard approach for software product planning and development used to be **requirements-driven**. This approach continues to be broadly used, for example for legal or regulatory requirements, commodity functionality requirements, and technology requirements. It is also applied in methods where hypotheses on customer needs, often referred to as "problems", and potential solutions are evaluated through qualitative and quantitative validation with customers, users and stakeholders.

Product planning in startups, in particular in B2C, is largely data driven:

1. **Data-analysis-driven** where the product team experiments with different implementations of design and product concepts, and makes decisions based on optimization objectives and the analysis of performance statistics or usage data. Data can also come from the (potential) user side covering how they behave in certain situations, or what their Jobs To Be Done (JTBD) are, or how much time they spend on what tasks. Such experimentation is a good approach when the focus is on innovation and optimization under uncertainty.
2. **Data-input-driven**. It applies to artificial intelligence/machine learning (ML) where data is used as input for an ML engine.

The choice of the software product planning and development approach is contextual, depending on the type of product, the target market, and the product scenario.

In the following sections, Customer Insight, Product Requirements Engineering, Release Planning, Roadmapping, and Product Life Cycle Management are considered primarily under a requirements-driven approach. All three approaches are covered in more detail in the ISPMA® syllabus “SPM - Excellence in Product Planning”.

Most of the tasks of product planning require the product manager to cooperate tightly with Development. 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 deliverables are managed.

Planning for and delivering the product in line with possible pivots (see 1.1) is a unique planning challenge for software product management in startups.

Literature: *Kittlaus, H.-B. (2020a); Kittlaus, H.-B., & Mangipudi, H. (2023); Blank, S. (2013, pp. 63-72); Bosch, J. (2009, 2019); Croll, A., & Yoskovitz, B. (2013); Fotrousi, F., & Fricker, S. (2016); Kittlaus, H.-B. (2012); Leffingwell, D. (2011); Mironov, R. (2008); Maalej, W. et al. (2015, p.33, p.48-56); Doorley, S et al. (2018); Stompff, G., & Smulders, F. (2015); Zacarias, D. (2019)*

3.2 Customer Insight

Duration: 0:30 h

Creating and evolving products that meet the ever changing needs of customers requires a clear understanding of the problems and the environment in which customers operate. Product managers must therefore work towards such an understanding. In this context, the term “customer” is used for all types of customer-side stakeholders, like user, buyer, IT manager, owner, operator, etc.

Two complementary approaches for gathering accurate customer data are available: Direct contact with customers and indirectly through the use of data analytics methods.

At a startup, it is important that as many roles as possible have direct contact with customers to understand their pain points and business requirements. However, the software product manager in particular plays an important role in representing the voice of the customer when conveying product feedback and scoping and refining feature requests.

In addition, startups typically do not have budget for externally contracted user testing and this responsibility may fall to the software product manager. This can include organizing product user groups as well as conducting product and design user testing and beta testing.

Product and design user testing includes (but is not limited to):

- When user testing should be conducted
- Who conducts/administers the testing
- How localization is handled (e.g. translations, prototypes)
- How testers are sourced and chosen
- How testers are incentivized
- Whether testing will be in person, remote, or both
- What equipment and tools are necessary for testing

Beta testing may also require a more formal process including:

- The number of beta testers and their geographic locations (e.g. for specific markets)
- Qualification requirements for sourcing and choosing beta testers
- Access to beta functionality is made available to testers for a specific amount of time and both quantitative and qualitative data are collected at the end of the testing period

Collecting relevant customer data is only the first step to create insights. Software product managers need to feed this data into discussions with stakeholders and use it for requirements analysis, as well as business modeling.

Startups may neither have the advantage of historical data of usage of the product nor the insights from usage of competitor's products. However the startups have the advantage of no baggage - a clean-slate approach of observing the target customers and examining the "Jobs to be done". And the experiences the customer is seeking from the whole-product helps the startups build a competitive differentiation. Thanks to the big data revolution, startups can now collect an enormous and diverse amount of customer information, with speed and granularity to perform analyses of it.

Secondly the data-driven approach should give contextual insights relating to the circumstances of the customers and their aspirations that can't be met by the current alternatives.

Literature: *Alvarez, C. (2014); Knapp, J., et al. (2016), Ries, E. (2011, Ch.1-4); Christensen, C.M. et al. (2016); Kittlaus, H.-B., & Mangipudi, H. (2023)*

3.3 Product Requirements Engineering

Duration: 1:00 h

Requirements engineering (RE) in a software product management context covers typical requirements engineering activities such as elicitation, analysis, specification, validation, and management, adapted to a market-driven situation with many customers, competitors, and suppliers.

Three requirement types can be distinguished and must be managed separately: Functional requirements, quality requirements, and constraints.

Literature: *Pohl, K., & Rupp, C. (2011); Regnell, B., & Brinkkemper, S. (2005); ISO/IEC 9126-1:2001 (2001); ISO/IEC 25010:2011 (2017); ISO/IEC 40500:2012 (2019)*

Requirements can be raised from different sources:

- Stakeholders
e.g. customers, users, user groups, business experts, executive management, partners
- Data sources
e.g. literature, social media, market analysis, product strategy, company guidelines, analytics, standards and regulations
- Systems in operation
e.g. other existing software products, competitive analysis

The goal of product RE is to continuously identify and manage requirements needed to implement the product strategy and address stakeholder needs. These requirements are called “product requirements” and cover requirements for the whole product (such as requirements for the business model, pricing, or marketing aspects).

Product requirements address no individual customer, but rather one or many markets which consist 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 software product manager’s responsibilities, and tightly linked to specific SPM activities needed in a market-driven context, such as creating a product vision, strategy definition, roadmapping, innovation, and release planning. Product requirements often result in requirements for development, but there are also product requirements that address other areas, e.g. bundling, new pricing models, new delivery processes, or improved support.

Literature: *Kittlaus, H.-B., & Mangipudi, H. (2023); Lucassen, G., et al. (2016, pp. 383-403); Regnell, B., & Brinkkemper, S. (2005)*

Needs and expectations arrive in different shapes and forms to product organizations and are named differently, depending on their abstraction level and development methodology. Different styles are used for documentation of requirements before development starts. Well known especially in agile development approaches are epics and user stories. A user story is a natural language description of one or more features of a software system from the perspective of the stakeholder. Epics often represent collections of user stories, sometimes referred to as “themes”. In agile development, requirements are documented with an approach to minimize waste and with a focus on the user experience.

Literature: *Gorschek, T., & Wohlin, C. (2006); Pohl, K., & Rupp, C. (2011); Regnell, B., & Brinkkemper, S. (2005); Agile Alliance (2020)*

3.4 Release Planning

Duration: 0:30 h

Traditionally, release planning is concerned with defining the detailed contents of a forthcoming product release in order to maximize the value of the release in relation to the product’s success over its life cycle. It needs to be tightly linked to the product requirements engineering process.

With agile methodology and continuous integration (and even continuous deployment) new functionality can ideally be delivered on a daily basis if customers are able to deal with such a high release frequency. This applies in particular in the Speedboat product scenario (see Figure 5). 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. One SPM challenge is to focus on the strategic and important items in such a high-pace environment. Another challenge is to communicate the added product value over time.

Software Product Scenarios		Life Cycle Phase	
		New Product Revolution	Existing Product Evolution
Runtime Environment	Vendor-Controlled	Powerboat	Speedboat
	Customer-Controlled	Icebreaker	Cruiseship

Fig. 5 Software Product Scenarios (H.-B. Kittlaus (2015))

The release planning process at startups tend to be very fluid for three reasons:

1. Because of the iterative cycles of Build-Measure-Learn which enhance the responsiveness to market in the pursuit of product-market fit.
2. Because of the interdependence with a multitude of ecosystem products.
3. As the startups gain traction their customer base gets more diverse – from enthusiasts and visionaries to broader market segments of pragmatists and more. In their pursuit of early product-market fit startups have to deliver these incremental benefits in quick time.

Therefore startups have much shorter release cycles compared to their established peers in the same class. The shorter release cycles also mean the startups have to design shorter and easier deployment and ensure fast delivery of the value benefits.

Literature: Berander, B., & Andrews, A. (2005); Carlshamre, P., et al. (2001); Cohn, M. (2006); Fricker, S., & Schumacher, S. (2012); Lehtola, L., et al. (2007); Ruhe, G. (2010); Wohlin, C., & Aurum, A. (2005)

3.5 Roadmapping

Duration: 0:30 h

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

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

With the roadmap, the product manager sets priorities and defines not only strategic content, but also budgets and resources for releases and versions. 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 for communication with customers, market research analysts, or investors. At a startup, making an external roadmap available to customers (or at least key customers), allows them to provide feedback and influence the development of the product. This both builds trust within the customer base and ensures that the roadmap decisions made early on in a product's life cycle are in line with market demand, thus strengthening the offering and probability of the continuation of the company. High-level release themes from the roadmap are key elements in guiding the release planning process.

The shorter release cycles also benefit the startups to keep the buzz in the market and the roadmaps help in building the stickiness.

Roadmaps also contribute to value selling in startups by articulating the benefits of the new version than just stating the features or benefits.

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

Literature: *Kostoff, R., & Schaller, R. (2001); Phaal, R., et al. (2004, 2007); Pichler, R. (2016, Part II)*

3.6 Product Life Cycle Management

Duration: 0:30 h

Product management is responsible for a product along the entire life cycle. Each phase of the life cycle has its individual characteristics and focus areas (Table 1). In the Startup phases, investments are necessary to develop the product.

Phase	Focus Areas
Conception and creation	Innovation, positioning, investment
Market Introduction	Launch, market share, investment
Growth	Market share, functionality, investment

Table 1: Software Product Life Cycle – Startup phases

While a company is still considered to be in a startup phase, the product (or products) typically are in the “Conception and creation”, “Market Introduction”, and “Growth” phases. Because of this, the software product manager involves many stakeholders in defining what will be built and why, how the product will be positioned and brought to market, and how it will gain market share and investment. For example, by involving Marketing early in the “Conception and creation” phase, a joint understanding and agreement can be reached as to why the product is being developed, i.e. what pain points does it solve and how will its messaging towards the markets be crafted.

Literature: *Anderson, C., Zeithaml, C. (1984); Rajlich, V. & Bennett, K. (2000); Moore, G.A (2014, Ch. 1-2)*

EU4 Strategic Management and Functional Areas

Duration: 2:45 h

Educational Objectives:

- EO4.0.1 Know the activities that belong to Corporate Strategy and Organizational Management
- EO4.0.2 Understand the role of a product management in supporting Corporate Strategy and Organizational Management
- EO4.0.3 Know the core tasks of the functional area Development
- EO4.0.4 Understand the different kinds of architecture and their importance to SPM
- EO4.0.5 Know the cooperation requirements between SPM and Marketing
- EO4.0.6 Know the cooperation requirements between SPM and Sales and Fulfillment
- EO4.0.7 Know the cooperation requirements between SPM and Delivery Services and Support

4.1 Corporate Strategy And Organizational Management Essentials

Duration: 0:15 h

Corporate Strategy is an activity within the startup aiming to define, plan, agree, implement and evaluate the organizational strategy. Founders / Executive Management's primary responsibility is to build the startup from concept into a viable, scalable and profitable business. While Software Product Management focuses on readiness of the product that fits the market needs, Founders / Executive Management ensure the organizational readiness – i.e. adequate funding, developing and honing of talent, external stakeholder management, staffing and building the organizational functions / teams like Development, Marketing, Sales and Support. Ensuring the compliance with all the applicable laws of the countries the company operates in is an important responsibility of the Founders/Executive Management often delegated to Product Management.

Literature: *Kim, W.C., & Mauborgne, R. (2015); Mintzberg, H., et al. (2008); Porter, M.E. (1998)*

4.2 Competitive Strategy

Duration: 0:15 h

As a challenger to the incumbent market players, Startup's Founders / Executive Management have relentless focus on their competitive strategies – i.e., creation of value differentiation or Unique Value Proposition (UVP) and creation of robust moats. Moats are a strong line of defense that prevent potential competitors and market forces from posing a risk to a new business's profitability, operational efficiency, and outlook for success. Examples of moats include Intellectual Property, a large and loyal customer base, a robust, non-replicable, cost effective product creation and delivery model, exclusive partnerships in the ecosystem, wide and cost-effective delivery channels. A network of moats

help the startup to present robust and long lasting defenses even as they are searching the path to profitability and sustainability.

Product Portfolio as a Moat: Inorganic expansion of product and/or market footprint is common in high growth startups and scale-ups. Product Management function plays an important role both in the pre and post acquisition process for the strategic acquisitions – predominantly in value amplification.

Innovation as a Moat: Startups’ survival depends on competitive differentiation and for that they employ both inventions as well as innovations.

Invention is the first occurrence of an idea for a new product or process, while innovation is the first attempt to carry the idea out in practice. Therefore inventions (and associated patents) by themselves may not help the cause of a startup unless they “use” such inventions and turn them into innovations that create compelling value for the customers.

Literature: *Kim, W.C., & Mauborgne, R. (2015); Kittlaus, H.-B., & Mangipudi, H. (2023); Porter, M.E. (1998) Chesbrough, H.W. (2005); Gorschek, T., et al. (2010); Ries, E. (2011, Ch.1-4)*

4.3 Funding

Duration: 0:15 h

Fundraising is important for survival as well as success of startups. Since time is of essence for fundraising, it is necessary to convince investors why it is the right time for them to get on-board with the given startup. For this, the startup needs to tell them the significance of the product, its transformative power and cultural fit with the changing market context. Prospective as well as current investors need crisp transparent and timely inputs on the following:

- Market and Business Outlook
- Plans vs. Progress
- Indicative timelines for product / market milestones – to facilitate their investment calls

Lastly, in startups, investor relations are not an exercise of one-way communications, but an active engagement to leverage the investors on multiple fronts beyond the financial commitments. Startups’ criteria for the choice of investors goes beyond the financial investment commitments and valuations - startups prefer investors who can bring in complementary assets like larger market access, complementary product / service offerings to enhance the competitiveness.

For new product development projects in mature organizations, a budget is given to the project team, often in combination with a stage-gate process, which requires the team to demonstrate frequently that its progress justifies additional funding.

Cashflow is an important metric for the startups' survival as well as growth. The term cashflow refers to the net amount of cash and cash equivalents being transferred in and out of a company. Investing cashflow in the context of product management refers to the cash used for investing in the product development which is expected to generate returns in the future.

Literature: *Fisher, L.M. (1998); Quinn, M. (2011)*

4.4 Compliance Management

Duration: 0:15 h

Compliance means the act of obeying an order, rule, or request (Oxford Dictionary), in more detail:

On the legal side: implementing any relevant legal or regulatory requirements

On the non-legal side: acting in accordance with any relevant external or internal standards and guidelines, e.g. in the areas of sustainability or ethics

Compliance management means the management of the decision process, including which legal and regulatory requirements are relevant, and which non-legal standards and guidelines the organization wants to comply with. It also includes a governance approach that ensures that the defined compliance requirements are consistently implemented and audited in the organization. It may include participation in and/or influencing of defining external and internal rules, standards and guidelines.

For sustainability, guidelines such as the United Nations 17 Goals and/or the Sustainability Awareness Framework may be considered.

Wherever applicable, Company guidelines from compliance management have to be considered in software product definition and may lead to additional product requirements. The alignment of company guidelines and the software product is in the responsibility of the software product management.

It is not uncommon to see innovation a step ahead of regulation. Therefore, in addition to complying with the existing regulations, innovative startups and new products need to be aware of evolving regulatory and compliance norms in their space and plan their compliance.

Literature: *United Nations, (2015); Duboc L., et al. (2019); Greenspan, A. (2002); Kittlaus, H.-B., & Mangipudi, H. (2023)*

4.5 Market Analysis

Duration: 0:15 h

It is of utmost importance for a startup 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 for the product level where the product manager needs reliable information.

Market need assessment is of top focus for early stage startups in their search for “venturable” problems. Customers’ propensity to pay is an important element of market analysis as the startups survey the alternatives the customers could choose from. It is common in startups for the software product managers to hold the responsibility of Market Analysis and providing actionable insights to the founders / executive management.

Literature: *Kittlaus, H.-B., & Mangipudi, H. (2023)*

4.6 Initiation and Management of Functional Areas

The division of work in a company that develops software products usually implies that there are separate functional areas for development, marketing, sales, consulting, services and support, each with 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. 2). Neither the framework nor the following descriptions give recommendations or make assumptions about the mapping of these functional areas to a company’s organizational structure. Early stage and growth startups may not have a full-fledged functional organizational structure, which requires the product management function to shore up those responsibilities initially, and help develop those independent functions as the organization scales up.

4.6.1 Development

Duration: 0:30 h

Development is responsible for all technical software aspects, including the implementation of changes and extensions to the software.

Product Architecture Management is done by the leading technical architect of a software product. 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 tight cooperation between product manager and architect. This table suggests a split of responsibilities. The SPM parts belong to the Product Strategy (see chapter 2).

Architecture dimension	Software Product Manager (SPM)	Technical architect (TA)
Offering architecture SPM Lead	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) SPM Lead	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 TA Lead	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 TA Lead	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 TA Lead	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 2: Software Architecture Dimensions

Business Architecture needs to be defined by modeling experts in cooperation with domain experts. We assume that they are part of the SPM organization.

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 architect must pay attention to the continuous improvement and protection of the defining technology of their software product.

Development Environment Management addresses development aspects that are relevant across and above development projects. This includes the governance of development processes and tools, IT infrastructure, configuration management, knowledge management, resource and skills management, development sourcing, and estimations.

Development Execution addresses the execution of the actual software development work. How this is done depends on the chosen development methodology, which in particular, has an impact on the way SPM and Development cooperate (see section 3.1). Development may work based on a project

structure, or in a continuous mode. Development is usually responsible for writing internal documentation and contributing to the software-related external documentation.

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. 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 into account 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 such as ISO 9241.

DevOps 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. It is primarily applicable to the Speedboat scenario (see section 3.1). At the core of a DevOps setup is a collaborative culture.

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

Product Management in some cases may take up user acceptance responsibility, to validate the implementation of a given requirement. The customer proximity and functional competence are potential reasons for such alignment. As the authors of functional requirements, product management is in a good position to validate the requirements without an elaborate and structured documentation.

Literature: *Bass, L., et al. (2015); Forsgren, J., et al. (2018); Hassenzahl, M., & Tractinsky, N. (2006); McGrath, M. (2000, Chapter 3); ISO/AWI TR 9241-1 (1998); Kittlaus, H.-B. (2012); Kittlaus, H.-B. (2022, p. 247 ff); Lazzaro, N. (2004); P. Le Callet, et al. (2013); Shneiderman et al. (2013); Jabbari, R., et al. (2016); Luz, W.P. et al, (2018); Leite, L., et al. (2019, Surv. 52, 6, Article 127); Erich, F., et al. (2017); Skelton, M., & Pais, M. (2019, Ch.1-3)*

4.6.2 Marketing

Duration: 0:30 h

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. The actual split of responsibilities between Marketing and Sales may differ from company to company. In some companies, SPM and Marketing are organizationally combined. At startups, the software product manager is typically a key contributor to marketing-related tasks, in particular product marketing. These product marketing efforts can include working together with the marketing team to create sales collateral and the messaging used to convey the positioning of the product which is then reflected on the company website and advertising campaigns. In addition, the software product manager may often be called upon to provide subject matter expertise to produce

the content for marketing material such as blogs, ebooks, and whitepapers generated by the marketing team.

With marketing efforts being lean and focused in a startup situation, the product roadmap and planned releases must be in sync with the marketing campaign plan to ensure a unified approach and shared efforts by both teams to achieve all aligned efforts.

Channel optimization means the selection, implementation, and management of channels appropriate for a product. The management of marketing partners within the product ecosystem can be part of the marketing responsibility unless there is a dedicated partner management organization. Both require tight cooperation with SPM and alignment with the corresponding product strategy.

Value communication is the process of connecting defined customer values with identified target markets for the product. This is done via relevant communication and engagement tools conveying the value messaging in formats, content, and channels adapted to the customer's buyer journey.

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 from existing and 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. With limited resources available at a startup, the software product manager is involved in discussions with market research analysts and press, and represents the product knowledge and strategy, alongside the C-level.

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. At a startup, the software product manager may be directly responsible for channel preparation or share that responsibility with Marketing and Sales. This typically includes creating product-specific training material, hosting it in a Learning Management System, and administering it to all stakeholders. As the primary champion of customers inside the startup, Product Management needs to help Marketing with the broader canvas of end-to-end Customer Experience with the startup. Optimizing the overall interaction management beyond the software touch points and ensuring customer stickiness through superior customer experience are critical for making the business model work.

Literature: *Bech, H.P. (2015); Kittlaus, H.-B. (2022, p. 256 ff); Sandy, K. (2020, Ch.2 & 10); Anderson, J.C., et al. (2007, Ch.2-3); Osterwalder, A. (2014); Elis, S., & Brown, B. (2017, Ch.1 & 10); Weber, L. (2020, p75ff & p100ff)*

4.6.3 Sales and Fulfillment

Duration: 0:15 h

The Sales function is responsible for all sales activities at a company. The actual split of responsibilities between Marketing and Sales may differ from company to company. Fulfillment means making the product available to the customer for use.

Customer relationship management means the systematic management of a company's interactions with customers 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.

Early stage startups may need product management to evangelize the product in the target markets / segments and establish the value-selling tools for the sales team.

Literature: *Kittlaus, H.-B. (2022, p. 265 ff); Brown, M., & Ellis, S. (2017); Sandy, K. (2020, Ch.2 & 10); Anderson, J.C., et al. (2007, Ch.2-3); Osterwalder, A. (2014); Weber, L. (2020, p106ff)*

4.6.4 Delivery Services and Support

Duration: 0:15 h

Delivery Services mean all customer-specific services provided to customers to help them become productive with the initial software product or when a new version is installed.

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.

Technical support refers to the fulfillment of contractual obligations, i.e. maintenance contracts with license products or of service-related elements of SaaS contracts. The typical support structure is:

Level 1: Help Desk

Level 2: Technical Maintenance

Level 3: Change Team (typically in or with Development)

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

Early stage startups may need their product managers to participate in the helpdesk / technical support to get a first hand feel of customers' experience to complete the feedback loop.

Operations is a key element in all products offered with the delivery model of SaaS (Software as a Service) or as a customer-specific managed service. In these cases, the vendor assumes the responsibility of operating the software at an internal or external data center, also known as a hosting

service, and giving access to the customer. The quality requirements of this hosting service are defined in a service level agreement that is part of the contract with the customer (see section 2.9). The vendor's approach may include DevOps (see section 4.6.1).

Product-related documentation is required both internally and externally. Internal documentation includes documents such as specifications, technical manuals etc. This documentation is usually developed by Development and 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, other ecosystem actors 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.

Early stage startups may need product management to set up the skeletal operations to fulfill the contractual and service level agreements. Secondly, product management's involvement in the early stage of setting up the operations helps the startups with an invaluable feedback loop to fine-tune the customer experience (CX) design as well as the business model.

Literature: *Fricker, S., et al. (2010); Kittlaus, H.-B., & Mangipudi, H. (2023); Raiffa, H., et al. (2007); Thompson, L. (2014); Lemay, M. (2017, Ch.2, 12); Sandy, K. (2020, Ch.2); Pichler, R. (2020, p.17ff, p.28ff, p.44ff, p.59ff, & p.74ff); Skelton, M., & Pais, M. (2019, Ch.7); Lemay, M. (2017); Kowalkowski, C., & Ulaga, W. (2017, Ch.1 & 9)*

Bibliography

Recommended SPM Overviews:

Altman, S. (2016): Before Growth, samaltman.com

Areitio, A. (2019): From Idea to Scale-up: Startup stages, Venturecity.

Kittlaus, H.-B., & Mangipudi, H. (2023): Software Product Management for Startups, InnoTivum/guNaka

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

van de Weerd, I., Brinkkemper, S., Nieuwenhuis, R., 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.

Almquist, E., Senior, J., Bloch, N. (2016): The Elements of Value. Harvard Business Review

Alvarez, C. (2014): Lean Customer Development – Build Products Your Customers Will Buy. O’Reilly

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

Anderson, J.C., Kumar, N., & Narus, J.A. (2007): Value Merchants

Andreesen, M. (2007), "The Only Thing That Matters ". pmarchive.com

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

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

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

Blank, S. (2013): Why the Lean Startup Changes Everything. Harvard Business Review, vol. 91, no. 5

Blank, S. and Dorf, B. (2012, p. xvii): "The Startup Owner’s Manual – The Step-By-Step Guide for Building A Great Company". K & S Ranch Inc.

Blank, S. (2014): "How Investors Make Better Decisions – The Investment Readiness Level ", steveblank.com

Borg, M., Chatzipetrou, P., Wnuk, K., Alégroth, E., Gorschek, T., Papatheocharous, E., Shah, S.M.A., Axelsson, J. (2019): Selecting component sourcing options: A survey of software engineering’s broader make-or-buy decisions. Information and Software Technology 112

Bosch J. (2019): *How to develop software*. LinkedIn, <https://www.linkedin.com/pulse/how-develop-software-jan-bosch/>

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

Camuffo, A. Cordova, A. and Gambardella, A. (2017): *A Scientific Approach to Entrepreneurial Decision-Making: Evidence from A Randomized Control Trial*, Discussion Paper DP12421 Centre for Economic Policy Research

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

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

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

Christensen, C.M et al. (2016): *Competing Against Luck – The Story of Innovation And Customer Choice*. HarperCollins

Clements, P.; & Northrop, L. (2015), *Software Product Lines: Practices and Patterns*. Addison Wesley.

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

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

Croll, A., Yoskovitz, B. (2013): *Lean Analytics – Use Data to Build a Better Startup Faster*. O'Reilly

Cusumano, M., Gawer, A., Yoffie, D.B. (2019): *The Business of Platforms*. Harper Business

Cusumano, M. (2004): *The Business of Software*. Free Press

Doorley, S., Holcomb, S., Klebahn, P., Segovia, K., & Utley, J. (2018): *Design Thinking Bootleg*. Stanford University

Duboc L., et al. (2019), *Do we really know what we are building? IEEE 27th International Requirements Engineering Conference (RE)*. IEEE

Dunford, A. (2019): *Obviously Awesome – How to Nail Product Positioning so Customers Get It, Buy It, Love It*. Ambient Press

Ebert, C., Brinkkemper, S. (2014): *Software product management - An Industry Evaluation*. *Journal of Systems and Software*, vol. 95

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

Elis, S., & Brown, B. (2017): *Hacking Growth*

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

Fisher, L. M. (1998): *Inside Dell Computer Corporation: Managing working capital*. *Strategy + Business*, accessed June 15, 2023, from www.strategy-business.com/article/9571

- Forsgren, N., Jumble, J., Kim, J. (2018): Accelerate - The Science of Lean Software and DevOps: Building and Scaling High Performing Technology Organization. IT Revolution Press*
- Fotrousi, F., Fricker, S. (2016): Software Analytics for Planning Product Evolution. International Conference on Software Business (ICSOB 2016), Ljubljana, Slovenia*
- Fricker, S., & Schumacher, S. (2012): "Release Planning with Feature Trees: Industrial Case", 18th Intl. Working Conference on Requirements Engineering. Foundation for Software Quality (RefsQ 2012), Essen, Germany*
- Fricker, S. (2012): Software Product Management, in Mädche A., Botzenhardt, A., & Neer L. (eds.): Software for People - Fundamentals, Trends, and Best Practices. Springer.*
- Fricker, S., Gorschek, T., Byman, C., & Schmidle, A. (2010): "Handshaking with Implementation Proposals: Negotiating Requirements Understanding". IEEE Software 27(2):72-80*
- Gorschek, T., Fricker, S., Palm, K., & Kunsman, S. (2010): "A Lightweight Innovation Process for Software-Intensive Product Development". IEEE Software 27(1):37-45*
- Gorschek, T., Wohlin, C. (2006): "Requirements Abstraction Model". Requirements Engineering 11:79-101*
- Greenspan A. (2002) : Regulation, innovation and wealth creation. Federal Reserve*
- Haines, S. (2014): The Product Manager's Desk Reference. McGraw-Hill Education Ltd*
- Hassenzahl, M., & Tractinsky, N. (2006): User Experience – A Research Agenda. Behavior & Information Technology 25(2):91-97*
- Hehn, J. Uebernickel, F. (2018) : "The Use of Design Thinking for Requirements Engineering ". 26th IEEE International Requirements Engineering Conference*
- Iansiti, M., & Levien, R. (2004): The Keystone Advantage – What the New Dynamics of Business Ecosystems Mean for Strategy, Innovation, and Sustainability. Harvard Business School Press*
- Ingenbleek, P. T. M., Frambach, R. T. and Verhallen, T. M. M. (2013) : "Best practices for new product pricing: Impact on market performance and price level under different conditions". Journal of Product Innovation Management, Vol. 30 No. 3, pp. 560-573.*
- ISO/AWI TR 9241-1 (1998): Ergonomics of Human-System Interaction. Part 1: Introduction to the ISO 9241 Series*
- 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*
- Jansen, S., Cusumano, M., & Brinkkemper, S. (2013): Software Ecosystems: Analyzing and Managing Business Networks in the Software Industry. Edward Elgar Publishers*
- Jansen, S., Peeters, S., & Brinkkemper, S. (2013): Software Ecosystems: From Software Product Management to Software Platform Management – From Start-up to SaaS Conglomerate. Life Cycles of Software Products (IWLCSP'13)*
- Khurum, M., Khurum, A., & Gorschek, T. (2007): A Model for Early Requirements Triage and Selection (MERTS) Utilizing Product Line Strategies. 11th International Software Product Line Conference, Kyoto, pp. 97-104.*
- Kim, W.C., & Mauborgne, R. (2015): Blue Ocean Strategy: How to Create Uncontested Market Space and Make the Competition Irrelevant. Extended Edition. Harvard Business School Press*

Kittlaus, H.-B. (2020b): Platform – What Do We Mean? <https://www.linkedin.com/pulse/platform-what-do-we-mean-hans-bernd-kittlaus/>

Kittlaus, H.-B. (2020a): Increasing Diversity in Software Product Planning and Development Approaches. <https://www.linkedin.com/pulse/increasing-diversity-software-product-planning-hans-bernd-kittlaus/>

Kittlaus, H.-B. (2019): Customer-Specific Tailorability of Your B2B Software Product – the Good, the Bad and the Ugly. <https://www.linkedin.com/pulse/customer-specific-tailorability-your-b2b-software-product-kittlaus/>

Kittlaus, H.-B. (2012): Software Product Management and Agile Development, in A. Mädche, A. Botzenhardt, L. Neer (eds.): Software for People - Fundamentals, Trends, and Best Practices. Springer

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

Klemens, B. (2006): Math You Can't Use – Patents, Copyright, and Software. Brookings

Knapp, J., Zeratsky, J., & Kowitz, B. (2016): Sprint: How To Solve Big Problems and Test New Ideas in Just Five Days. Transworld Digital

Kostoff, R., & Schaller, R. (2001): Science and Technology Roadmaps. IEEE Transactions on Engineering Management 48(2):132-143

Kowalkowski, C., & Ulaga, W. (2017): Service Strategy in Action. Service Strategy Press

Lazzaro, N. (2004): Why We Play Games: Four Keys to More Emotion Without Story. Report. XEO Design

Le Callet, P., Böller, S., Perkins, A. 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

Leffingwell, D. (2011): Agile Software Requirements. Addison-Wesley

Lehtola, L., Kauppinen, M., & Vähäniitty, J. (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

Leite, L., Rocha, C., Kon, F., Milojić, D., & Meirelles, P. (2019): A Survey of DevOps Concepts and Challenges. ACM Comput. Surv. 52, 6, Article 127

Lemay, M. (2017): Product Management in Practice. O'Reilly UK Ltd

Lucassen, G., Dalpiaz, F., van der Werf, J., Brinkkemper, S. (2016): Improving Agile Requirements: The Quality User Story Framework and Tool. Requirements Engineering, vol. 21, no. 3

Luz, W.P., Pinto, G., & Bonifácio, R. (2018): Building a Collaborative Culture: A Grounded Theory of Well Succeeded DevOps Adoption in Practice. ACM/IEEE International Symposium on Empirical Software Engineering and Measurement

Maalej, W., Nayebi, M., Johann, T., Ruhe, G. (2015): Toward Data-Driven Requirements Engineering. IEEE Software

Maurya, A. (2012): Running lean: Iterate from plan A to a plan that works. Sebastopol, CA: O'Reilly

Maurya, A. (2013), Why not the Funnel Chart. goodreads.com

McGrath, M. (2000): Product Strategy for High-Technology Companies – Accelerating Your Business to Web Speed (2nd ed.). McGraw-Hill

- Messerschmitt, D.G., Szyperski, C. (2003): *Software Ecosystem – Understanding an Indispensable Technology and Industry*. MIT Press
- Mintzberg, H., Ahlstrand, B., & Lampel, J. (2008): *Strategy Safari: The Complete guide through the wilds of strategic management (2nd ed.)*. Financial Times Prentice Hall
- Moore, G.A. (2014): *Crossing the Chasm – Marketing and Selling disruptive products to mainstream customers*. HarperBusiness
- Moore, G.A. (2005): *Inside the Tornado – Strategies for Developing, Leveraging and Surviving Hypergrowth Markets*. HarperBusiness
- Nagle, T.T., Hogan, J.E., Zale, J. (2014): *The Strategy and Tactics of Pricing – A Guide to Growing More Profitably (5th ed.)*. Pearson Prentice Hall
- Nagle, T.T., & Hogan, J.E. (2005): *What Is Strategic Pricing? Strategic Pricing Group Insights (Monitor Group)*
- Osterwalder A., & Pigneur Y. (2010): *Business Model Generation*. Wiley
- Osterwalder, A. (2014): *Value Proposition Design*. Wiley
- Parker, G.G., Van Alstyne, M.W., Choudary, S.P. (2016): *Platform Revolution*. W.W. Norton
- Phaal, R., 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
- Phaal, R., Farrukh, C., & Probert, D. (2004): *Technology Roadmapping – A Planning Framework for Evolution and Revolution*. *Technological Forecasting and Social Change* 71:5-26
- Pichler, R. (2020): *How to Lead in Product Management - Practices to Align Stakeholders, Guide Development Teams, and Create Value Together*. Pichler Consulting
- Pichler, R. (2016): *Strategize: Product Strategy and Product Roadmap Practices for the Digital Age*. Pichler Consulting
- Pohl, K., Rupp, C. (2011): *Requirements Engineering Fundamentals*. Rocky Nook Computing
- Popp, K.M. (2015): *Best Practices for Commercial Use of Open Source Software*. Books on Demand
- Popp, K.M., & Meyer, R. (2010): *Profit from Software Ecosystems: Business Models, Ecosystems and Partnerships in the Software Industry*. Books on Demand
- Porter, M.E. (1998): *Competitive Strategy*. Free Press
- Pritchard, C.L. (2013): *Risk Management: Concepts and Guidance (5th ed.)*. CRC Press
- Quinn, M. (2011): *Forget about profit, cash flow is king*. *Wall Street Journal*, accessed June 15, 2023, from <http://online.wsj.com/article/SB10001424052970204524604576609740825745286.html>
- Radcliff, A. (2013): *Why you should find product-market fit before sniffing for venture money*. *FastCompany*
- Raiffa, H., Richardson, J., & Metcalfe, D. (2007): *Negotiation Analysis – The Science and Art of Collaborative Decision Making*. The Belknap Press of Harvard University Press.
- Rajlich, V. & Bennett, K. (2000): *A Staged Model for the Software Life Cycle*. *IEEE Computer* 33(7):66-71
- Riani, A. (2019): *The Five Biggest Stages of A Startup From Idea to Scale*. *Forbes*

- Regnell, B., & Brinkkemper, S. (2005): *Market-Driven Requirements Engineering for Software Products*. In Aurum, A., & Wohlin, C. (eds.): *Engineering and Managing Software Requirements*. Springer
- Ries, E. (2011): *The Lean Startup – How Today’s Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses*. Penguin
- Ruhe, G. (2010): *Product Release Planning: Methods, Tools and Applications*. CRC Press
- Sandy, K (2020): *The Influential Product Manager*. Berrett-Koehler Publishers
- Schmidt, M. (2002): *The Business Case Guide*. Solution Matrix
- Schwaber, K., Sutherland, J. (2020): *The Scrum Guide - The Definitive Guide to Scrum: The Rules of the Game*. <https://www.scrumguides.org>
- Schwager, A and Meyer, C (2007): *Understanding Customer Experience*. Harvard Business Review
- Shneiderman, B., Plaisant, C., Cohen, M., Jacobs, S. (2013): *Designing the User Interface: Strategies for Effective Human-Computer Interaction (5th ed.)*. Addison-Wesley
- Skelton, M., & Pais, M. (2019): *Team Topologies: Organizing Business and Technology Teams for Fast Flow*. It Revolution Press
- Sodhi, M.N., & Sodhi, N.N. (2007): *Six Sigma Pricing: Improving Pricing Operations to Increase Profit*. Pearson FT Press
- Stompff, G. and Smulders, F. (2015), *The Right Fidelity: Representations That Speed Up Innovation Processes*. 19th DMI: Academic Design Management Conference
- Thompson, L. (2014): *The Mind and Heart of the Negotiator (6th ed.)*. Pearson Prentice Hall
- Torres, T. (2021): *Continuous Discovery Habits*. Product Talk LLC
- United Nations, (2015): *The 17 Goals*. https://sdgs.un.org/#goal_section
- Weber, L. (2020): *Software Product Management – Tips and Templates*, Amazon
- Wohlin, C., & Aurum, A. (2005): *What is Important when Deciding to Include a Software Requirement in a Project or Release?* International Symposium on Empirical Software Engineering
- Zacarias, D. (2019): *20 Product Prioritization Techniques*. Folding Burritos
- Zowghi, D., Coulin, C. (2005): *Requirements Elicitation: A Survey of Techniques, Approaches, and Tools*. In Aurum, A., & Wohlin, C. (eds.): *Engineering and Managing Software Requirements*. Springer