1 Introduction

Why do we write another book on software product management (SPM)? And what is so special about product management for software compared to other products?

Product management is a discipline that has been utilized by many industries for decades, above all the consumer goods industry. The invention of product management as an explicit management concept is attributed to Procter and Gamble. In 1931, the company assigned one product manager to each of two competing soap products (see [Gorche11]). Since then, this basic idea has become widespread. In fact, it makes sense for any company to manage explicitly the products that generate its revenue and that, as assets, represent the company's sustainable value. But what does product management mean? Unfortunately, a general answer can only be given for parts of this question. The activities of the product manager depend largely on the type of product involved, the culture, history, and organization of the company, and the target and reward system. Product management means planning and coordinating all relevant areas of a product inside and outside the company with the aim of sustainably optimizing product success.

The focus of this book are the tasks of a software product manager, and how they can be performed well. We follow a best-practice approach and make use of any techniques that help product managers, from whichever methodology they come, be it lean, agile, kanban etc. In our experience, methodologies come and go over time, the tasks remain. Therefore, we do not use temporarily marketable terms like "Agile Product Management" or "Lean Product Management". The prime objective of a product manager is not to follow any particular fashionable methodology, but to take care of his tasks in the best possible way, and thereby make his product successful over its life cycle.

Software has become highly pervasive. There is hardly any industry that is not increasingly dependent on software, be it as part of their products or as the backbone of their business operations. A good example of that phenomenon is the automobile industry. The software contents of cars have increased significantly over the last 10 years. In more and more industries software is turning into the number one value driver.

H.-B. Kittlaus, S.A. Fricker, *Software Product Management*,
DOI 10.1007/978-3-642-55140-6_1

Another example is corporate IT organizations. For them, it has become apparent that they need to manage their software applications explicitly as crucial assets with a life cycle perspective, i.e. as software products. The realization that companies from non-IT industries are suddenly becoming “standard software suppliers” by providing apps to their customers may also have contributed to this development.

Software has become important also in our private sphere. Many use software to manage their private lives and to communicate with others. Software has also started to be critical for addressing the big challenges of our societies. For example in the healthcare area, internet-based apps empower patients and their caregivers to make decisions for themselves. This empowerment allows expert healthcare staff to serve more patients for less cost per patient.

Software product management—as we define it—does apply to all these organizations:

- Vendors of software products and software-intensive technical services, be it licensed products, or Platform-as-a-Service (PaaS) or Software-as-a-Service (SaaS) including apps running on all kinds of smart devices (software industry),
- Vendors of software-intensive products (all industries), e.g. cars or smartphones,
- Vendors of professional human services (all industries) in which software is used to increase productivity, and
- Corporate IT organizations (all industries).

In several decades of experience in the software industry, the authors have come to realize that the knowledge and experience acquired in other business areas and with other types of products are only partially transferable to software. We believe that software is the most complex product of human invention that we know of (see Chap. 2). That complexity implies that the management of software products is special and puts unique demands on the persons responsible for this task (see also Sect. 2.6).

A software product manager’s job is special because software is special, i.e. the characteristics of software are different from most other products and have a strong influence on what a software product manager does. Most conspicuous are

- High frequency of change over the life cycle of the software with the resulting great importance of requirements management.
- High complexity.
- Ability to interact with customers through the product.
- Flexibility of re-configuring the product and adapting it to new purposes and usage contexts.
- Culture of searching for lightweight, agile approaches to building and evolving software.
- No or little need for physical manufacturing and distribution.
- Increasing returns through network effects.
- Special financial picture due to low marginal cost.

When Hans-Bernd Kittlaus published his first book on Software Product Management [KiRaSch04] and his second book on “Software Product Management and Pricing” [KittClou09], there were not too many publications on this subject available. Since then, the situation has changed significantly. The number of SPM-related publications has increased, there is more focused research in academia, and there is an increasing number of commercial SPM training offerings available.

These changes are to some degree due to the establishment of International Software Product Management Association (ISPMA, www.ispma.org). First convened in 2009, ISPMA is a non-profit organization whose fellow members are SPM experts from the industry and academia. Other types of membership are open to all interested companies and individuals. ISPMA’s goal is to foster software product management excellence across industries (for more information see Sect. 7.4). ISPMA’s fellows have developed a curriculum and a Certifiable Body of Knowledge (SPMBOK) that have become the basis for a high number of commercial training offerings and university courses. Samuel Fricker and Hans-Bernd Kittlaus are co-founders of ISPMA. Hans-Bernd is ISPMA’s current chairman, Samuel is ISPMA’s former chairman and current board member.

1.1 About this Book

This book is intended to provide an integrated view of software product management for all the organizations and scenarios listed above. We consider the similarities between the scenarios that are induced by the specifics of software as more significant than differences in organizational views. Nevertheless, there are some differences that require different priorities regarding the individual tasks (see Sect. 7.3).

The target group of this book is everyone involved or interested in software product management, be it on the academic side or in organizations in the scenarios listed above.

With this book, the authors intend to provide a state-of-the-art update of the SPM part of [KittClou09] that is compliant with ISPMA’s Body of Knowledge[1] (as of December 2016):

- ISPMA SPM Foundation Level Syllabus V.1.3
- ISPMA SPM Excellence Level Syllabus Product Strategy V.1.1
- ISPMA SPM Excellence Level Syllabus Product Planning V.1.1
- ISPMA SPM Excellence Level Syllabus Strategic Management V.1.2
- ISPMA SPM Excellence Level Syllabus Orchestration V.1.0.

[1] www.ispma.org

As a result, the reader can use the book for supporting the preparation of an examination for ISPMA certification. However, the primary source for preparation is always the current release of the corresponding syllabus and the corresponding training.

We introduce the ISPMA software product management (SPM) framework in Chap. 2. The remaining chapters will follow that structure. The subject of software pricing will be addressed in this book (see Sect. 3.10), but not as extensively as in [KittClou09].

Due to the wealth of publications related to SPM over the last couple of years, the authors do not claim to provide a comprehensive bibliography. We have rather restricted ourselves to publications most often cited and publications we find particularly useful. By reading this book, the reader will know the views of the most influential results of research aimed at SPM topics.

Clear boundaries are needed to be able to address the breadth of the topic of software product management. Despite important interfaces between development and software product management, neither the topics of software development management nor project management will be discussed here. We have also omitted operations and consulting issues: the book does not cover bookkeeping, administration, and payment for the software licenses acquired and used in a company. Also, it does not cover software product line management, i.e. the management of development variants based on the same product platform.

The book is structured as follows. Chapter 2 looks at software as a business and how we suggest to manage it in terms of business models and organizational aspects. We define relevant terms and introduce the ISPMA SPM Framework. Chapter 3 describes the tasks and activities related to product strategy. Chapter 4 focuses on product planning. Chapter 5 covers strategic management. Chapter 6 describes the orchestration of the organization's functional areas. Chapter 7 rounds things up by looking at SPM's state-of-practice, the application of SPM in different business scenarios, as well as the future of SPM.

At the end of the book, we provide a glossary, i.e. a list of definitions of all relevant terms used. It is aligned with ISPMA's glossary as of December 2016.

1.2 Conventions

Before we begin our in-depth discussion of SPM, we need to clarify several conventions used throughout this book. Terms such as "manager" or "director" are meant to be gender-neutral, that is they refer equally to female and male persons. While women increasingly participate in IT management, we have chosen a convention of referring to such positions using the male pronoun. This is not intended to be in any way discriminatory and was chosen simply for the purposes of easier readability. We use terms like development or marketing with small characters when we mean the activity, with capital characters when we mean the organizational unit.

We use the term “software vendor” when we mean companies whose primary business is the development and provision of software for commercial purposes for a relatively large number of consumer and business customers. Examples are Microsoft, SAP, Oracle, IBM, Google, Samsung and Huawei. We use the term “corporate IT organization” for organizational units that are part of companies in all industries and whose primary mission is to provide software or IT support for the parent company or corporation, which we call “corporate customer”. In that sense, software vendors are corporate customers of their internal IT.

The term “service” has many different meanings (see Webster’s Dictionary). The following three meanings are relevant to this book:

- Useful labor that does not produce a tangible commodity (as in “professional services”).
- A provision for maintenance and repair (as in “software maintenance service”).
- The technical provision of a function through a software component that can be accessed by another software component. Such access often occurs over a network and executed on a remote server (as in “web services”, “Software-as-a-Service”, or “Service-Oriented Architecture”).

Whenever we use the term “service” in this book, we try to make it clear which meaning is intended.

When we use the term “controlling” we mean “performing the functions of a (business or financial) controller”.

And now, we can jump right into the secrets of software product management.