

# Concepts of Product Software: Paving the Road for Urgently Needed Research <sup>\*</sup>

Lai Xu and Sjaak Brinkkemper

Institute of Information and Computing Sciences,  
Utrecht University, 3584 CH Utrecht, The Netherlands  
emails:{L.Xu,S.Brinkkemper}@cs.uu.nl

**Abstract.** Software products are an everyday phenomenon. Yet, there are very few scientific studies reported on the engineering of software as a product in particular. This paper stipulates the urgent need for more research on product software. The various terms used for software products are reviewed and some categorizations of software products are presented. Moreover, we provide a software product development framework to position the key business domains in a product software company.

## 1 Introduction

Product software accounts for substantial economic activity all over the world [1], [2]. In 2001 (1999) the total market of the product software industry was estimated to be 196 (154.9) billion USD, which is just 9% of the overall ICT spending of 2.1 trillion USD worldwide. “The product software sector is among the most rapidly growing sectors in OECD countries, with strong increases in added value, employment and R&D investments.” [2]. Although this percentage of product software usage differs from country to country, the trend in many organizations is that the make-or-buy decision falls more and more in favor of purchasing of a standard software product.

Despite the economic importance of product software, there is still very limited research activity on the development of software as a product. Generalizing experiences from product development a body of knowledge needs to be established with theories, methods, and tools. According to our observations, we are still at the beginning of a long journey.

Looking at earlier work, some research has been focused on software processes in small-size firms [3] which includes some research in portfolio management [4], [5] and software design quality improvement [6] for small software firms. The main purposes of their research has been to report on a preliminary investigation of in small firms which have different business strategies, organization issues and product development processes from other kinds of firms.

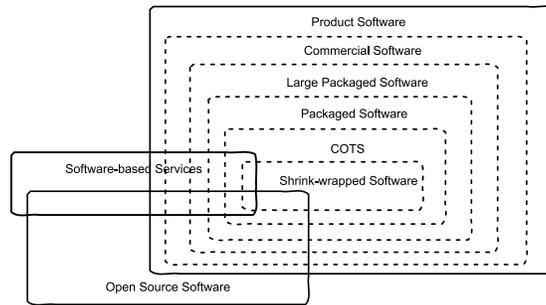
---

<sup>\*</sup> This work is sponsored by the Netherlands national research organization NWO, Jacquard project Overture (project number: 638.001.203).

This paper is organized as follows. First, we present the various terms, provide our definition of software products and discuss specifics of the software business in Section 2. In Section 3, a software product development framework is presented. We conclude with some remarks on future work in Section 4.

## 2 What are software products?

When we talk about software product, terms like shrink-wrapped software, commercial off-the-shelf (COTS) software, packaged software and commercial software immediately come to mind. Other concepts like open source software and application service provision (ASP) should also be considered. In common literature [7], [8], [9], [10], the boundaries distinguishing the concepts of shrink-wrapped, COTS, packaged and commercial software are blurred. Relationships between these types of software are presented in Fig. 1.



**Fig. 1.** Software products

*Shrink-wrapped software* is software on mediums that are boxed, shrink-wrapped and sold in stores. Shrink-wrapped software also implies a widely supported standard platform.

*COTS software* is developed for a whole market instead of individual customers. A COTS product, such as an application or a component, is sold, leased, or licensed to the general public; offered by a vendor trying to profit from it; supported and evolved by the vendor, who retains the intellectual property rights; available in multiple, identical copies and used without source code modification [11].

*Packaged software* describes ready-made software products that can be readily obtained from software vendors, and which generally require little modification or customization. The term today typically refers to upscale enterprise software suites, such as ERP (Enterprise Resource Planning) systems. These packaged software, although ready-made, rarely comes ready-to-run. *Large packaged*

*software* typically requires weeks or months of deployment and implementation work to set it up for the specific needs of each individual business.

*Commercial software* is software which is purchased through the retail market. Commercial software must be bought or licensed before using. Making copies of this software without the express permission of the author or controlling party is usually prohibited.

*Open source software* is software for which the underlying “source” code is readily available for inspection and modification by any interested person. This contrasts with most commercial software, for which the source code is a closely guarded trade secret. Most open source software has some type of license agreement for its use that may cover rights to modify, redistribute, use for commercial purposes, and so on. Open source software is not necessarily free in price, redistribution is generally allowed though.

*Application service provisions* (ASPs) also offer application software that runs on the servers of the providers. Some of them are free to use; most of them are not.

Given all those different opinions and terms on software products, it is important to standardize views and establish one common definition. We therefore propose the following definition:

*A **software product** is defined as a packaged configuration of software components or a software-based service, with auxiliary materials, which is released for and traded in a specific market.*

In this definition, ‘packaged components’ refer to all software discussed above which implies code, executables and web pages; ‘software-based services’ cover concepts like ASPs sold commercial software services; ‘auxiliary materials’ consists of software documentation, user manuals, training material, brochures and the like; and ‘release and trading’ identifies their commercial value.

## 2.1 Categories of software products

Other software product classifications originate from two different perspectives: standard/ semi-standard, and from the Organization for Economic Co-operation and Development(OECD).

Software products can be classified according to architectural standards and language standards, and other standards/semi-standards. For example, from the architectural standards view, possible architectural patterns are centralized, client-server, 2-tier, 3-tier, peer-to-peer, pipe and filter and blackboard etc [12].

According to the OECD, software products can be categorized as system infrastructure software, software development tools, or application software [2]. “System infrastructure software” includes system level operating system and other software, middleware, system management, and security software. “Software development tools” contains database management systems, development environments, development life-cycle management, and internet tools. “Application software” covers ERP systems, cross-industry business software, CAD /CAM/CAE and other vertical industry business software.

## 2.2 What is specific to software business?

The software business is a special industry where making one copy or one million copies of software product costs about the same [13]. Software investments can result in substantial productivity gain and strategic advantage [14], like in the movie, music and medicine production industries. It is also a business with up to 99% gross profit margins for its product sales.

In the software business, productivity of the best employee and the worst one has frequently up to ten- or twenty fold difference. About 75% to 80% of the product-development projects are also late and over budget [13].

Customers of software business are easier “locked in” to a particular vendor because of product decisions some one made a decade or two ago that can not easily be reversed [13].

These and other specific issues in the product software industry call for designated research effort.

## 3 Software product development framework

Software product development can be viewed from organization and development perspectives as shown in Fig. 2. From the organization perspective, a product software firm will develop its corporate strategy, product strategy and service strategy according to its target market. The product and service strategies are in service of the corporate strategy. Process and quality control provides an overview to the production process and guarantee certain level of quality.

When software product firms create their corporate strategy, internationalization can be seen as the key prerequisite for continued growth in the product software business [15]. In the software business, product investment typically includes a choice of standardized products for mass markets. The internationalization and product investment is especially relevant for the resource management where technology development has traditionally been the central area of interest among managers.

The product strategy considers the product lifecycle management and portfolio analysis. Mapping market trends to software product design is becoming the central issue in software product development [16], [17].

The service strategy is aimed at improving service quality. For large size software products such as ERP software, product software firms can choose between building customized software systems by themselves or to outsource the customization project to vendors that perform the complete implementation activities [13], [18]. Those decisions eventually affect the service strategy of product software firms. The process and quality control improves effective engineering work and brings substantial quality.

From the development perspective, the product software includes requirements engineering, architecture/design development, delivery and implementation services stages. Requirements management includes capturing market trend, analyzing requirements and releasing the software product at right time.

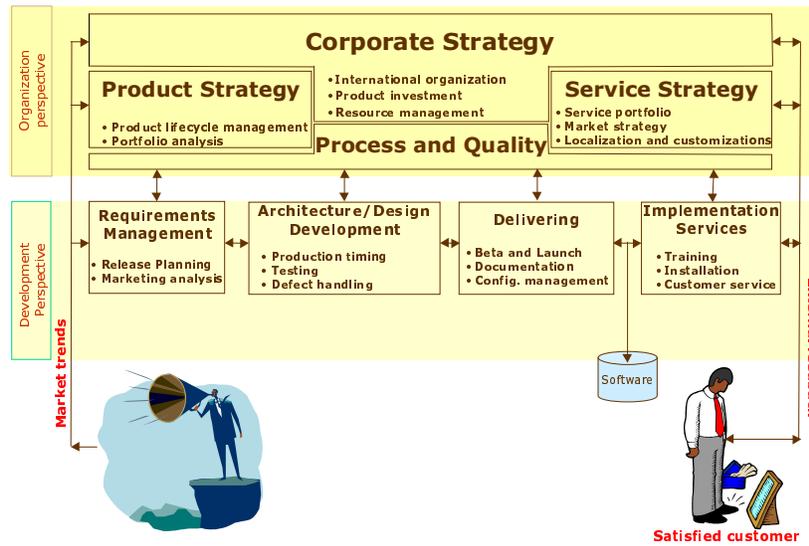


Fig. 2. Software product development framework

Software architecture is fundamental for the development of software products. Work in software architecture can be seen as attempting to codify the structural commonality among a series of software products so that the high-level design decisions inherent in each product need not be re-invented, revalidated and re-described[12].

Delivery is an important step at which software products are launched on and offered to the market. Typically delivery consist of configuration management and documentation. The goal of software configuration management is to keep evolving software products under control and help satisfy delay and quality constraints. Software documentation describes the requirements of software products which need to be satisfied, the design, implementation, capabilities and limitations of the software product to make the product easier to use, maintain and reuse.

After more than 30 years of software development, most software applications, such as large-scale information systems, are constructed by adapting existing software products. Implementing software product into customer systems, such as training, installation and custom services, can be done by the software product company or be outsourced this part business to consulting firms. Both choices exist at the current time [13].

## 4 Conclusions

In this paper, we have placed some first stones on the road for research in the area of product software. We have reviewed software product related concepts and provided our definition of a software product. A brief overview of some of the categorizations for software products has been provided. Moreover, we presented the specific characters of software product. The software product development framework provides a positioning of the research focus for both the organizational and the development perspective of a software product company.

In our future research, we will focus on software product development of all types of software product firms and consider relationships between all business strategies and software development stages.

## References

1. OECD: The software sector: Growth, structure and policy issues. OECD Report DSTI/ICCP/IE(2000)8/REV2 (2001)
2. OECD: Highlights of the oecd information technology outlook 2002. OECD Report (2002)
3. Rautiainen, K., Lassenius, C., Sulonen, R.: 4cc: A framework for managing software product development. *Engineering Management Journal* **14** (2002) 27–32
4. Vähäniitty, J., K., R.: Towards an approach for portfolio management in small product-oriented software companies. In: *Proceedings of Hawaii International Conference on System Sciences (HICSS-38)*, IEEE Computer Society (2005) 1–10
5. Vähäniitty, J.: Product portfolio management in small software product businesses - a tentative research agenda. In: *Proceedings of the 6th International Workshop on Economic-Driven Software Engineering Research (EDSER-6) at ICSE 2004*. (2004)
6. Mäntylä, M.V.: Developing new approaches for software design quality improvement based on subjective evaluations. In: *Proceedings of the International Conference on Software Engineering (doctoral symposium)*. (2004) 48–50
7. Sawyer, S.: Packaged software: Implications of the differences from custom approaches to software development. *European Journal of Information System* **9** (2000) 47–58
8. Carmel, E., S.Sawyer: Packaged software development teams: What makes them different? *Information Technology & People* **11** (1998) 7–19
9. Sawyer, S.: A market-based perspective on information systems development. *Communications of the ACM* **44** (2001) 97–102
10. Sawyer, S., Guinan, P.J.: Software development: Processes and performance. *IBM system Journal* **37** (1998) 552–569
11. Brownsword, L., Oberndorf, T., C.Sledge: Developing new processes for cots-based system. *IEEE Software* **17** (2000) 83–86
12. Bass, L., Clements, P., Kazman, R.: *Software Architecture in Practice*. Addison Wesley Longman (1998)
13. Cusumano, M.A.: *The Business of Software*. Free Press (2004)
14. Brynjolfsson, E.: The productivity paradox of information technology. *Communication of the ACM* **36** (1993) 66–77
15. Alajoutsijarvi, K., Mannermaa, K., Tikkanen, H.: Customer relationships and the small software firm a framework for understanding challenges faced in marketing. *Information and Management* **37** (2000) 153–159
16. Hauser, J., Clausing, D.: The house of quality. *Harvard Business Review* **66** (1988) 63–73
17. Garvin, D.A.: *Managing Quality: the Strategic and Competitive Edge*. Free Press (1998)
18. Nelson, P., Richmond, W., Seidmann, A.: Two dimensions of software acquisition. *Communication ACM* **39** (1996) 29–35