Association for Information Systems
AIS Electronic Library (AISeL)

PACIS 2016 Proceedings

Pacific Asia Conference on Information Systems (PACIS)

Summer 6-27-2016

# ECONOMICS OF SOFTWARE DEVELOPMENT

Wenge Zhu
*California State Polytechnic University*, zhu@cpp.edu

Yuanjie He
*California State Polytechnic University*, he@cpp.edu

# ECONOMICS OF SOFTWARE DEVELOPMENT

Wenge Zhu, College of Business Administration, California State Polytechnic University, Pomona, CA, USA, zhu@cpp.edu

Yuanjie He, College of Business Administration, California State Polytechnic University, Pomona, CA, USA, he@cpp.edu

## Abstract

*In this paper, we investigate economics of software development. We first identify several dimensions of software quality from user's perspectives. We model the tradeoff between having more features and functionalities and the ease of learning in usability, and the tradeoffs among ease of use, ease of learning, and usage frequency. Based on the impacts of quality dimensions on user's utility, we analyze the software monopoly's economic incentives to provide high quality software products and build analytical model to analyze the software company's optimal design decisions.* *(Keywords: Software, Usability, Economics)*

# 1. INTRODUCTION

Software industry has been one of the fastest growing sectors, the research on managerial decision in software development, however, is not very rich. Software is a typical information product, or development intensive product as studied in Krishnan and Zhu (2006), meaning that besides the cost of sales and promotion, the cost of software is almost solely the cost from its development. Hence, product development in software industry is even more crucial than NPD in most other industries.

Since the nature of the software products is complex, the learning curve for the software products can be steep. It is well known that the installation and implementation of SAP ERP is very costly (Wailgum 2009). Beside the fixed learning cost, there is usually a variable usage cost and variable usage benefit associated with software products. The software users' perception on the value of software products depends on the tension between these factors, namely, learning cost and net usage benefits. The usage benefits are largely related to the features and functionalities of the software products, and are also related to the user interface design. On the other hand, the software development companies have the tension between providing more features and functionalities and keeping the user interface simple. To address these tradeoffs, in this paper, we build analytical models to describe and analyze the tensions and conflicts of interests in order to provide strategic guidance in software development.

This paper focuses on the economics of the software development and investigates the development strategies for the monopoly company. First, we review the relevant literature in general R&D and software development. Then we investigate the software quality and its different dimensions. Based on usage related quality model, we study the monopoly inefficiency in the NPD in software products, and the incentives for the monopoly to provide software quality.

## 1.1. R&D Economics Literature

The first research in R&D can date back to Schumpeter (1939), who indicates the relationship of inventive activity and investment & output growth in his theory of business cycle. In his theory, the inventive activity is considered exogenous. Schmookler (1962) is the first to include the inventive activity endogenous in his framework. The economic literature in R&D mostly concerns the interaction between R&D and investment, and the technological opportunity is considered the central role in that interaction (Lach and Schankerman 1989). It is generally believed that R&D is determined by both the pure economic factors, such as input price and demand condition, and the advance in the science and technology (Rosenberg 1969, 1974; Pakes and Schankerman 1984; Griliches, Hall, and Pakes 1988). Institutional roles in R&D are also widely recognized through organizational structure (Tompson 2012, Lee and Walsh 2016) or merge and acquisition (Frankort 2016, Rindfleisch 2001).

The existing literature on economics in R&D normally takes the traditional game theory modeling approach. The competitions among the firms lead to an R&D race, which is modeled as either a tournament one (only one firm can be successful) as in Canbolat et al (2012), or non-tournament one (possibility of multiple successful firms in the end of the game) as in Poyago-Theotoky (1999).

The limit of the mainstream economic R&D research with regard to its implication in management research is obvious. First, it does not distinguish between the research process and the new product development process - two processes that are set with different goals and are judged by different criteria even though they are closely related. Second, it does not clearly differentiate process innovation and product innovation. Process innovation is normally the result of continuous improvement of related processes to the existing product and is usually modeled as cost reduction effort; while product innovation is normally a breakthrough innovation and is characterized by the introduction of a new product.

A related stream of research is the research on NPD in business literature. It typically addresses only the "D" part in R&D and includes more operational factors, such as the process and the organization of NPD projects, costs related to operational details. Readers may refer to the review papers by Krishnan and Ulrich (2001) and Krishnan and Loch (2005).

### 1.2. Software Development Literature

The research in software development in management literature has two important areas: the software development project management, and the customer requirement capturing.

The first topic is mainly about quality control in software development (Farr 1996, Slaughter et al 1998, Harter et al 2000, Axelsson 2016) and cost estimation (Kemerer 1987, Mukhopadhyay and Kekre 1992, Maxwell et al 1999, Hu et al 1998, and Huang et al 2015), and most of the work is done empirically with emphasis on the organization of software development projects, levels of management involvement, etc.

The second topic is about the nature of customer requirement and customer satisfaction, and the matching between them and the software product (Kekre et al 95). General customer satisfaction on software products can be categorized and measured in the following seven factors (Kekre et al 95, Narayanan et al 2011) – reliability, capacity, usability, installability, maintainability, performance, and documentation. Another description of the customer requirement is by Varian (1993). He uses two different user costs to characterize the software quality (or customer satisfaction): fixed cost of learning, and variable cost of operating the software. From the above literature, we have the following observations:

- Most of the R&D literature is limited for managerial insight into software NPD.
- Most of the software NPD research is empirical; there are very few established analytical models, especially the models which characterize the uniqueness of software and other knowledge products.
- Most of the software NPD research is implicit about the type of the software studied.

There is an apparent gap between the economic literature on R&D and the literature on software development: economic literature is rarely industry-specific and therefore their models fail to consider the nature of the software products, managers may find the findings in economic literature too generic to be helpful; on the other hand, the software development literature focus too much on engineering details and typically never consider the factors on business side, such as marketing. Our paper tries to fill this gap by providing an analytical framework in the research of software NPD by incorporating both the software-specific operational factors and users' perspective in our economic model.

## 2. MODEL AND ASSUMPTIONS

To capture the major issues in customer satisfaction when developing new software product, we rearrange the seven factors categories of customer satisfaction in Kekre et al (95) into three major ones:

1. Utility: it is identical to the Kerkre et al's definition of capacity;
2. Fixed usage cost: it incorporate the learning effort in usability, and the installability;
3. Variable usage cost: it incorporates the operating effort in usability, the maintainability, as well as the performance. The three parts are combined to reflect the common characteristic that they are in proportion of the usage.

The significant of the above categorization of software quality dimensions is that it is evolved around a central theme of consumer usage, and is thus ready to be used in managerial decisions, such as market segmentation. We can characterize the consumer profile by his or her potential software usage frequency. Before introduction to our model, we list the notations we will use in our model:

| | |
|---|---|
| $x$: | The number of potential buyers of the software. |
| $p$: | The price charged for the software. |
| $v$: | The variable cost of the software for the user, including the cost of operating the software each time, the cost of maintaining the software each time, we call it variable usage cost. It measures part of the usability, maintainability, as well as the performance. |
| $b$: | the utility provided by the software per usage, usually in term of the number of features in the software. We will use this measurement of utility to model the economic benefits from using the software each time. |
| $F(b)$: | The fixed cost of the software for the user, including the cost of learning the software, of installing the software. We call it fixed usage cost, which measures part of the usability and installability. This cost is positively related to the utility of the software since higher utility implies more functionalities and features and more functionalities and features in turn imply higher complexity. Therefore, $F(b)$ is monotone increasing in $b$, or $F'(b) > 0$. |
| $k$: | The fixed cost of producing (including the fixed cost from copying, streaming, and/or distributing) the software. |
| $c_v(v)$: | The cost of designing software with variable usage cost of $v$. |
| $c_F(F, b)$: | The cost of designing software with fixed usage cost of F and utility of $b$. |
| $c_x(x)$: | The variable cost of manufacturing $x$ copies of software, including production, sales and distribution. |
| $c_b(b)$: | The cost of designing software with usage benefit of $b$ |
| $n$: | The number of usage of the software in a given period. |
| $g(n)$: | Density function of consumer distribution, or the number of consumers using the software n times often |
| $S$: | Net Surplus of consumers. |
| $U$: | Consumer's objective function. |
| $P$: | Producer's objective function. |

For simplicity, we assume that the cost of design and producing the software is a linear combination of three kinds of costs $c(x,b,v,F) = c_v(v) + c_b(b) + c_F(F,b) + c_x(x)$. We assume that the potential usage frequency of the software for any consumer is exogenously given by his or her personal information such as the nature of the job, the education background, etc., and thus is relatively fixed and will not be changed from the usage of software. We can see that under our setting, the utility of a customer with usage frequency of $n$ is $(b - v) \cdot n - F - p$.

We assume consumers are heterogeneous in their software usage frequency. In order for a user to choose the software, the user's utility needs to be non-negative, which requires a minimum usage frequency of the software. We define a marginal user as a consumer with usage frequency of $n^*$ such that his/her utility is $(b - v) \cdot n^* - F - p = 0$. Therefore the marginal usage frequency is $n^* = (p+F)/(b-v)$. In another word, consumers with usage frequency higher than $n^*$ are willing to purchase the software and consumers with usage frequency lower than $n^*$ will not buy.

The firm will optimize its objective function $P$,

$$Max_{p,b,F,v}(p - c_x) \cdot \int_{n^*}^{\infty} g(t)dt - c_v(v) - c_F(F,b) - k,$$

Where $n^* = (p+F)/(b-v)$.

Let us now look at the consumer's net surplus. Let us denote $H(t)$ as the inversion of $G(n)$, where $x = \int_n^{\infty} G(t)dt$ is the number of potential buyers having at least the usage frequency of $n$. We have the net consumer surplus as below.

$$S = u(x) - p(x)x = \int_0^x [(b-v)H(t) - F]dt - [(b-v)H(x) - F]x$$
$$= H(x)(b-v) \cdot \left[\int_0^x H(t)dt - H(x)\right] = (b-v) \cdot \left[\int_0^x H(t)dt - \frac{p+f}{b-v}\right]$$

where $x = \int_{n^*}^{\infty} g(t)dt$ and $n^* = (p+F)/(b-v)$.

### 2.1. Model Analysis

In the following sections, we would like to further explore the following research questions.
1. For a software company offering single product, how ease of use and ease of learning separately affect consumers' surplus and how they jointly affect consumers' surplus?
2. If the software company may offer a product line consisting of higher and lower versions, what would be the companies' optimal decision on ease of use and easer of learning and whether product line improve consumers' surplus?
3. If the software company will offer a line of products and the company can develop the low-end version by stripping down the feature and functionalities at almost zero cost, is this practice optimal?

## Reference:


1. Axelsson, Jakob. (2016). Quality assurance in software ecosystems: A systematic literature mapping and research agenda. Journal of Systems and Software. 114. 69-81.
2. Canbolat, Pelin G.; Golany, Boaz; Mund, Inbal; Rothblum, Uriel G. (2012). A Stochastic Competitive R&D Race Where 'Winner Takes All'. Operations Research. 60(3). 700-715.
3. Cohen, Wesley M., Daniel A. Levinthal. (1990). Absorptive Capability: A New Perspective on Learning and Innovation. Administrative Quarterly. 35(1). 128-152.
4. Farr, W. (1996). Software reliability modeling survey. M. R. Lyu, ed. Handbook of Software Reliability Engineering. McGraw-Hill, New York. 71-115.
5. Frankort, Hans T. W. (2016). When Does Knowledge Acquisition in R&D Alliances Increase New Product Development? The Moderating Roles of Technological Relatedness and Product-Market Competition. Research Policy. 45(1). 291-302
6. Harter, D.E., M.S. Krishnan, S.A. Slaughter. (2000). Effects of process maturity on quality, cycle time and effort in software product development. Management Science. 46(4). 451-466.
7. Hu, Qing, Plant, Robert T., Hertz, David B. (1998). An empirical analysis of data preprocessing for machine learning-based software cost estimation. Journal of Management Information Systems. 15 (1), 143-163.
8. Huang, Jianglin; Li, Yan-Fu; Xie, Min. (2015). An empirical analysis of data preprocessing for machine learning-based software cost estimation. Information & Software Technology. 67. 108-127.
9. Kemerer, C. F. 1987. An empirical validation of software cost estimation models. Comm. ACM 30(5) 416-429.
10. Krishnan, M. S., C. H. Kriebel, S. Kekre, T. Mukhopadhyay. (2000). An empirical analysis of productivity and quality in software products. Management Science. 46(6) 745-759.
11. Krishnan, Viswanathan, Karl T. Ulrich. (2001). Product Development Decisions: A Review of the Literature. Management Science. 47 (1). 1-21.

12. Krishnan, V., Christoph H. Loch. (2005). A Retrospective Look at Production and Operations Management Articles on New Product Development. Production and Operations Management. 14 (4). 433-441.
13. Krishnan, Vish, Wenge Zhu. (2006). Designing a family of development-intensive products. Management Science. 52 (6). 813-825.
14. Lee, You-Na; Walsh, John P. (2016). Inventing While You Work: Knowledge, Non-R&D Learning and Innovation. Research Policy. 45(1). 345-59
15. Maxwell, K., Luk V. Wassenhove, S. Dutta. (1999). Performance evaluation of general and company specific models in software development effort estimation. Management Science. 45. n (June). 787-903.
16. Pakes, Ariel, Schankerman, Mark. (1984). An exploration into the determinants of research intensity. In R&D, Patents, and Productivity. Edited by Zvi Griliches. Chicago: University of Chicago Press (for NBER).
17. Narayanan S, Balasubramanian S, Swaminathan. J. (2011). Managing Outsourced Software Projects: An Analysis of Project Performance and Customer Satisfaction. Production & Operations Management. 20(4). 508-521
18. Poyago-Theotoky, Joanna. (1999). A Note on Endogenous Spillovers in a Non-tournament R&D Duopoly. Review of Industrial Organization. 15(3). 253-62.
19. Rindfleisch, Aric. (2001). The Acquisition and Utilization of Information in New Product Alliances: A Strength-of-Ties Perspective. Journal of Marketing. 65(2). 1-18.
20. Shapiro, Carl, Hal R. Varian. (1998). Information Rules: A Strategic Guide to the Network Economy. Harvard Business Review Press.
21. Slaughter, S. A., D. E. Harter, M. S. Krishnan. (1998). Evaluating the cost of software quality. Comm. ACM. 41(8). 67-73.
22. Thompson, Peter. (2012). The Relationship between Unit Cost and Cumulative Quantity and the Evidence for Organizational Learning-by-Doing. Journal of Economic Perspectives, 26(3). 203-24
23. Wailgum, Thomas. (2009). Want to Save $10 Million or More on ERP? Don't Buy Oracle or SAP. CIO. Feb 26, 2009.