Hybrid Business Models in Software Product Industry: Patterns and Challanges

Swanand J. Deodhar[1], Kulbhushan C. Saxena[1], and Mikko Ruohonen[2]

1 Fortune Institute of International Business, Vasant Vihar, New Delhi, 110057, India

2 CIRCMI, Department of Computer Sciences, University of Tampere, Kalevantie 4, 33014, Tampere, Finland

Abstract. Operating in software product industry is becoming an increasingly risky proposition. Compressed timeline for product development combined with need to reduce cost has compelled organizations to look at new ways of doing business. One such avenue is combining the erstwhile conflicting practices of open source and closed source software. This industry paper highlights common patterns and challenges encountered in operationalizing such business models. The findings are based on a larger multiple case study research involving six such software products.

1   Introduction

Software product industry represents organizations that develop software products and build business models around these products. These business models are influenced by two dominant and contrasting licensing regimes, namely proprietary and open source licensing. These two approaches have spawned diverse mechanisms for software development and distribution.

Although the two approaches are highly contrasting, they have an ironic capability of complimenting each other's weaknesses. In proprietary approach, vendors are in control of development and distribution and therefore vendors have to bear the cost for the same. In return vendors are assured of larger chunk of economic rents as only they can sell software licenses. On the other hand, open source approach allows for faster development and distribution channel as large pool of voluntary developers would contribute the software code and place the same in public domain thus removing any practical chance of selling licenses. Clearly, if a proprietary software vendor can benefit from development and distribution practices of FOSS and vice versa.

Such attempts are increasingly becoming necessary with organizations' survival in software industry becoming increasingly uncertain. Established software product vendors are threatened by reduced barriers to entry. With software product innovations happening rapidly [1] and leadership positions in software product industry becoming fragile [2], software product industry is increasingly becoming a risky proposition with firms ending up in bankruptcy in a short span of time. For

example, from 1995 to 2007, exit rate in software product industry was three times that of pharmaceutical industry and two times that of hardware industry [1].

Hybrid business models provide one form of innovation adopted by organizations to survive in such troubled environment [3]. Such merging represents number of challenges. How can organizations combine two approaches that were designed to be incommensurable? And are there any common patterns in such hybridization? These questions were the starting point of the study

1.1   Hybrid Business Models

Before we explain challenges in operationalizing hybrid business models, it is important to elaborate on notion of hybrid business models. Hybrid business models form a special class of business models. Software product business models are often divided into four categories: product development, product distribution, revenue model, and maintenance [4]. A business model hence can be treated hybrid if one or more its dimensions combines practices from both open source and proprietary software ecosystems.

Commonly reported example is dual licensing model where software is released under multiple editions with each edition governed by different licensing norms [5]. However we observed that most organizations adopted a single licensing approach. The generic form of such business model can be described as follows.

The software was available under multiple editions, each edition governed by the same licensing norms but different in functionality and support. The low-cost edition (often termed as community edition) was available for free and provided what one respondent called *commodity functionality*. The professional edition on the other hand extended such commodity functionality through extensions and contractual agreements for service provision. Most organizations studied had developed partner networks for providing services. Partners also participated in developing specific extensions which were also packaged with professional editions or could be purchased separately. The professional edition could be bundled with such partner-developed and at time proprietary editions owing to the terms of the customized permissive licensing. Users could avail professional edition on the basis of subscription. In the next two sections, we describe common patterns and challenges in operationalizing such business models.

1.2   Research Methodology

Owing to the contextual adherence of the phenomenon, we chose case study as the research methodology. Case study was adopted as research methodology. In disciplines such as medicine and law, case study research has been the most favored mode of investigation while organizational and social sciences, case study research is gaining acceptance [6-8]. The increasing importance of case study research as methodology is rooted in its potential to expose phenomenon and researcher to each

other in myriad of empirical avenues that otherwise are not possible through positivistic paradigm.

Case study research is suitable under certain conditions only. According to [8], case study research is appropriate to investigate a phenomenon within its real-life context, when the investigator has little control over the events. As the goal of the study was to expose trends and challenges experienced by organizations in operationalizing hybrid business model, it was hence imperative to examine hybrid business model in its context. Therefore, a case study method was chosen as the most appropriate approach for this research.

Before any case study endeavor is undertaken, important considerations have to be handled. In the subsequent sections, treatment of these considerations in the context of the study is explained. First issue is about the number of cases. As outlined by [8], multiple case study approach is suitable for a theory development exercise where each case is a separate experiment in itself. Each case compares the theoretical understanding and the new empirical evidences. On the other hand, single case study is suitable when phenomenon requires studying a unique, critical or revelatory case. Because we were interested in pattern identification, multiple case study approach was considered as the ideal choice. The cases were chosen as per replication logic (all cases having a hybrid business model), to improve on external validity of the findings.

Second consideration in case-based research pertains to the case selection criteria. Explicit mention of case selection criteria is key indicator of rigor in case research [8]. Overarching selection criteria for this study were presence of a hybrid business model associated with a software product. The idea was to ensure the fit of the case with the research questions at hand.

Thirdly, case study protocol was used to guide data collection. We wanted to capture data on operationalization of hybrid business model. Hence the protocol was developed accordingly with most questions began with 'how' and 'why'. We used interviews as the primary data collection vehicle. It is considered particularly suitable for the interpretive case studies [9]. Due to geographical limitations, interviews were conducted and recorded through video conferencing. Later these interviews were transcribed. We also used data from the published news articles and social media platforms.

We began the analysis process with open coding. The interview transcripts were coded for either operational practice of hybrid business model or an indicator of challenge in doing so. Once the open coding was done, we tried to examine the linkage across codes to identify axial linkages across coded excerpts. The exercise ended up with a set of operational practices and challenges related to hybrid business models.

1.3   Summary of the Cases

A total of six cases were studied. The cases belonged to four different product categories. One of the cases was an enterprise resource planning software. Three

belonged to the content management system product category. One belonged to the business process management system while the last one was a customer relationship management system. The organizations were geographically diverse as well. Two cases were from mainland Europe, one was located in Scandinavian Europe, one was from South American continent, and two were headquartered in USA.

In all the cases, one of the top members of top management (preferably CEO) was interviewed. The idea was to get an overall description of the business model. To understand the operationalization, personnel from business development, and product engineering were also interviewed. As external resources such as community members play a key role in open source approach, personnel from community management were also interviewed. Lastly, to understand the distribution management, personnel responsible for managing partner network was interviewed.

In some cases, single person represented more than one functions. So he/she was contacted for collection data on all relevant dimensions. A total of 23 full-length interviews were conducted across 6 cases. The primary data was substantiated with secondary collected through secondary sources such as forum interactions between different stakeholders, product roadmap document, and release policy document.

1.4   Hybrid Business Models: Patterns

OSS is largely governed by two forms of licensing: permission and restrictive [10] (Lerner & Tirole, 2005). In all the cases, we observed *adoption of customized permissive licensing*. Such licensing allowed community members freedom to choose licensing of their respective contributions. As one of the respondents stated it acts as an incentive for commercial organizations participating in the community. The licenses however were customized to provide legal protection to the brand name of the product. Common permissive licenses that were used as basis included Mozilla Public License and Lesser GNU Public License.

Secondly, we observed *community to be made up of multiple segments*. Unlike the traditional notion of community as a uniform social collection of developers, hybrid business model fostered communities made up of business partners, customers of professional edition, customers of free edition, and partners of free edition. Each segment participated in development of product with distinct motivations.

Technological proximity to open source was another pattern observed. Most organizations studied did not start out with hybrid business model. It emerged later as suitable way of doing business. However, organizations already were technologically closer to open source than proprietary with software product being created using open source technologies. As stated by most of the interview respondents, migration to hybrid model was therefore a natural choice.

Phased release was another common feature. Organizations would often release their community edition before corresponding release of professional edition. This allowed the organizations to capture the user-feedback (for example, bug reports and

feature requests) which was often incorporated for the professional edition, along with few more extensions.

Finally, most organizations were attempting to *create ecosystem around their products*. Towards this, they had developed a customized development and distribution platform. These platforms provided infrastructure for developing and hosting extensions for sales. Customers could search specific extensions through enhanced search functionality. In other words, entire ecosystem of the product could be developed around such platforms.

1.5 Hybrid Business Models: Challenges

Operationalization of such business models however involved certain challenges. Owing to the dichotomy of mixing open source with proprietary, most of these challenges were dichotomous.

Software products were released in multiple editions. Some of these editions were freely available while others came at a cost. Intended purpose of each edition was defined. The community edition was meant to achieve a faster rate of diffusion and to gather customer's feedback before releasing the professional editions. Usually community editions were not meant to be used in mission critical applications. Accordingly, organizations had to devise functional coverage of each edition. This functional differentiation across editions had to be wide enough for community edition users to perceive professional edition as valuable and upgrade. On the other hand, it also had to be narrow enough for community edition to qualify as a usable piece of software. Naturally, a community edition that did not have critical pieces of functionality would not allow for intended faster rate of diffusion. We observed that attempts to resolve this challenge led to differences of opinions between community members and the organization. We term this challenge as an extends the concept of *selective revealing* [11].

We have already posited segmented structure of community as a outcome of hybrid business models. In such segmented communities, different segments participated for different motivations. The challenge was to coordinate product development and distribution across these segments. For example, partners with interests in community edition only could package the edition and under certain licensing terms could also release the same as their own product. As one of the respondents stated, this led to a fractured user-base where multiple versions of the same editions were floating around in the market. One may be tempted to install strong appropriation regimes and take control of the product development and distribution. However, this would significantly take away benefits of hybrid business model. We term this challenge as *segmented meshing* where different community segments need to be meshed into a single entity. We observed adoption of a customized permissive license as a common approach to tackle this situation. Such a license provided intellectual property protection for the brand of the software product but still allowed community members to add proprietary functionality.

1.6   Conclusion

The paper focuses on emergence of hybrid business model in software product industry. It also provides some patterns and challenges in operationalizing such business models. The findings are based on data collected for a larger empirical study carried out as first author's doctoral work. Authors hope that reported findings would act as quick-start guidelines for software product organizations to looking to adopt hybrid business models.

References

1. Li, S., Shang, J., & Slaughter, S. A. (2010). Why Do Software Firms Fail? Capabilities, Competitive Actions, and Firm Survival in the Software Industry from 1995 to 2007. *Information Systems Research, 21(3).*

2. Schmalensee, R. (2000). Antitrust Issues in Schumpeterian Industries. *American Economic Review,* 90(2), 192-196.

3. Campbell-Kelly, Martin & Garcia-Swartz, D.D. (2010). The Move to the Middle: Convergence of the Open-Source and Proprietary Software Industries. *International Journal of the Economics of Business,* 17(2), 223-252

4. Rajala, R., Rossi, M. and Tuunainen, V. (2003). A Framework for Analyzing Software Business Models. Proceedings of the *ECIS 2003, European Conference on Information Systems* - New Paradigms in Organizations, Markets and Society, June 18-22, 2003, Naples, Italy

5. Välimäki, M. (2002) Dual Licensing in Open Source Software Industry *Systemes d´Information et Management, 8(1),* 63-75

6. Lee, A. (1989). A Scientific Methodology for MIS Case Studies. *MIS Quarterly , 13* (1), 33-50.

7. Eisenhardt, K.M. (1989) Building Theories from Case Study Research, *Academy of Management Review, 32(4)*

8. Yin, R. (2009). Case Study Research: Design and Methods (4th Ed.). SAGE Publication

9. Walsham, G (1995) Interpretive case studies in IS research: nature and method, European *Journal of Information Systems, 4(2),* 74-81.

10. Lerner, J., & Tirole, J. (2005). The Scope of Open Source Licensing. *Journal of Law, Economics, and Organization, 21(1),* 20-56.

11. Henkel, J. (2006). Selective Revealing in Open Innovation Processes: The Case of Embedded Linux. *Research Policy, 35(7),* 953-969.