

Chapter 10

Open Source Reuse and Strategic Imperatives

Rahul De' and Ravi A. Rao

Abstract Free and Open Source software (FOSS) allows firms to gain strategic advantage by enabling business agility that is essential to compete in a hypercompetitive environment. Literature on the strategic role of IT indicates that contemporary firms need to constantly upgrade their IT capabilities and maintain flexible IT systems to remain competitive. We argue that FOSS lends itself well to an iterative process of capability development: providing firms with a strategic advantage through reuse of FOSS. We conduct a qualitative case study-based research of commercial firms to assess the strategic imperatives of reusing FOSS. Results indicate that the primary motivation for reusing FOSS is the ability to innovate through access to open source components along with their embedded knowledge, supported by independence from vendor lock-in. Specific advantages enjoyed by firms through the use of FOSS include flexibility, interoperability, stability, security and time-to-market.

10.1 Introduction

Software reuse has the potential to provide economic benefits through enhanced quality and productivity [20], as has been shown through research in the software engineering domain. But does software reuse also provide firms with strategic benefits? Most research related to software reuse for business purposes has focused on the economic value of the reuse [7] and on the strategy around how to reuse software [8]. In this chapter, we explore whether software reuse, specifically the reusable assets of open source, provides firms with strategic benefits.

R. De' (✉) • R.A. Rao
Indian Institute of Management Bangalore, Bangalore, India
e-mail: rahul@iimb.ernet.in; ravi.rao10@iimb.ernet.in

Software reuse is fundamental to Free and Open Source software (FOSS). By opening up the source code, FOSS enables its users to not only view and modify the code of the native application, but also the ability to reuse pieces of code and graft them onto new applications to address different requirements. FOSS has emerged as the most successful form of large-scale software reuse with a range of reusable assets available for deployment across multiple technology platforms.

Our definition of reuse includes absorbing complete FOSS products that are enhanced for specific business requirements, as well as deploying FOSS components that are reused as building blocks for developing custom applications. In addition, our definition of reuse includes reuse of the code, as well as the reuse of knowledge (technical and domain knowledge) that is closely associated with the code.

The research questions addressed by this chapter can be stated as:

- Do firms reuse FOSS to gain strategic advantage?
- What attributes of FOSS motivate IT managers to reuse FOSS?

10.2 Literature Review

Before we attempt to answer our research questions, we synthesize past research done pertaining to (a) Role of IT in firm strategy, (b) Software reuse capability provided by FOSS, and (c) Types of FOSS reuse adopted by firms.

10.2.1 *IT as a Strategic Resource*

Traditional research exploring the role of IT in obtaining strategic differentiation for firms has drawn from two streams of strategy literature originating from (a) Porter's five forces and Porter's value chain models, and (b) the resource-based view of the firm. The first stream of work is drawn from the theory of industrial organization (IO) and explores the structure of the industry in terms of five competitive forces [21, 22] including: bargaining power of buyers and suppliers, threats of new entrants, threat of substitute products and intensity of competition. The role of IT is seen as impacting these forces to provide firms with a competitive edge within the industry structure [18]. While this stream of research provides an explanation of how a firm is able to leverage IT for manipulating the competitive forces, it does not address the question of how the competitive advantage can be sustained – as competition can imitate these actions and nullify the advantage. The second stream of research, drawn from Chamberlainian economics, explores the role of firms endowed with heterogeneous resources and their ability to extract superior rents from these resource endowments [2]. The resource-based view (RBV) of the firm suggests that firms create organizational capabilities by assembling these resources to work together. Adopting RBV, IS researchers have explored the role of IT resources such as IT infrastructure; technical and managerial IT skills; and IT-enabled intangibles

such as know-how, culture, reputation etc. as a potential source of competitive advantage [1]. The net prescription of this school of research is to develop strategic IT applications to sustain competitive advantage over time; and to align IT with business strategy, structure and processes [12].

While these streams of research provide insights on how IT can be leveraged to gain competitive advantage, these frameworks are not sufficient to explain the role of IT differentiation in contemporary business environments [24]. Businesses today operate in a *hypercompetitive* environment [5], which is characterized by competition occurring in the form of a series of market disruptions aimed at nullifying any supernormal returns enjoyed by the incumbent leader [4]. In such environments, advantages from specific competitive moves are temporary and superior performance is derived from continuously recreating competitive advantages through innovative actions [24]. A firm's ability to rapidly generate these competitive moves thus becomes a key strategic imperative [5]. The strategic requirements for firms in a hypercompetitive environment include agility, continuous innovation, time-to-market and the timing of the competitive moves [24].

In sum, the underlying theory of how and why IT innovations provide a strategic advantage has changed with the changing landscape of the business environment [24]. The traditional view of IT differentiation was aimed at overcoming Porter's competitive forces [18] and to leverage a firm's heterogeneous resources. With business environments becoming hypercompetitive, the logic of IT differentiation shifted towards enhancing competitive agility [24]. Consequently, the source of IT differentiation shifts from strategic applications to enterprise IT capability and from alignment to embeddedness of IT in business strategy [24]. Sambamurthy et al. [25] examine the strategic role of IT on firm performance and suggest the influence of IT capabilities on firm performance through organizational capabilities and strategic processes. They stress the importance of developing capabilities that allow the firm to dynamically combine IT and business resources; and processes that allow them to combine knowledge, assets and resources to craft innovations. Thus, to leverage IT for strategic differentiation, mere access to strategic applications is no longer sufficient. To achieve sustainable competitive advantage, firms need to have a combination of entrepreneurial alertness that enables them to sense strategic opportunities and IT capabilities that enables them to develop innovative solutions. This co-evolution of strategic processes needs the constant assembling of IT capability in iterative loops, as firms evolve their competitive position [25].

Thus, in today's hypercompetitive environment, firms not only need to possess superior IT capability, they also need to possess the agility that enables them to constantly discover and develop new knowledge, assets and resources. We argue that free and open source software lends itself well to this iterative process of capability development. The reuse of the open source repository rich in ideas, knowledge, techniques and solutions provides firms the ability to constantly evolve their IT capabilities in tune with their strategic processes.

10.2.2 Software Reuse and Open Source Software

Software reuse is the process of using existing software code or knowledge rather than building new software systems [14]. Software reuse can be considered as opportunistic or planned [3], and reused either from internal or external sources. Systematic reuse in software requires the meticulous building of a reusable artifact repository [14] and needs substantial investment in time and effort to identify reusable code, and to build the necessary tools and repository [11]. Systematic software reuse originated as a form of reusing knowledge and code from internal projects with the aim of deriving economic benefits through enhanced quality and reduced effort [20]. In order to derive such economic benefits, commercial organizations invested in building repositories of reusable artifacts.

FOSS is considered to be an example of large scale software reuse with a wide range of reusable assets [3]. With the advancement of open source, a rich and extensive set of external repositories with a wide range of reusable assets is now available not just to open source developers, but to commercial organizations as well. Unlike corporate reusable repositories which are specifically built to store reusable artifacts, FOSS is a collection of independent initiatives representing a variety of technology innovations and approaches. Considering the substantial investment required for developing a reusable artifact repository, research has focused on the phenomenon of creating a large-scale FOSS repository. Code reuse is a form of knowledge reuse that is fundamental to innovation [11] and the types of reuse in open source include the reuse of algorithms and methods; lines of code; and software components that encapsulate functionality [27].

The quality of the reusable artifact is an important determinant of the extent of reuse. Developers contributing to open source use popularity as a proxy for quality rating of the reusable components [11]. The motive for these developers is not just that of cost reduction but also the access to knowledge to foster innovation [3]. However, research also indicates that open source developers work under severe time and resource constraints and subject themselves to self-inflicted pressure to release code early [11]. A potential drawback of this practice is the possibility of the software components in FOSS not being sufficiently abstracted for effective search and reuse.

Thus, the emergence of FOSS as a reusable repository has the impact of providing an external and large repository of assets for knowledge reuse. This provides the opportunity for commercial developers to reuse not only software but also best practices including ideas, knowledge, techniques and solutions [3]. However, searching FOSS components for reuse may not be easy, given the argument that FOSS components are not developed with the objective of reuse. Given the nature of open source reusable artifacts and the difficulty in searching them, its probable that the motivation for commercial firms to reuse open source goes beyond mere productivity benefits and include benefits derived from reusing knowledge embedded in them.

Empirical studies on the reuse of open source software components indicate a pattern of reuse ranging from a few blocks of code to entire products [9]. Further, there is evidence of commercial firms preferring to use open source components as-is without any significant modifications [16]. Extending these arguments, we classify the reuse of FOSS by commercial firms as a) direct replacement for commercial off-the-shelf products needing no modifications; b) as customized open source products considerably enhanced for individual firm requirements; or c) as completely crafted applications built from a plethora of open source and in-house software components. The choice of how firms deploy and use FOSS varies depending on these objectives. Firms may choose to use FOSS for various benefits such as economic advantages, quality and productivity gains, scalability, performance improvement, security, flexibility and agility.

As argued in Sect. 10.2.1, firms seeking IT differentiation, seek to develop IT capabilities and deploy them to gain agility and flexibility. Reuse of FOSS provides one such means for firms to constantly upgrade their IT capabilities. We argue that firms seeking differentiation through FOSS would seek gains beyond mere economic and productivity benefits and leverage FOSS to gain competitive advantage. Such firms will go beyond deployment of as-is FOSS products and tend to reuse best-of-breed FOSS components to craft customized IT systems. The extent to which a firm will modify and integrate diverse set of FOSS components will vary depending on their view of FOSS reuse as a strategic differentiator.

10.3 Research Methodology

We used a qualitative approach to study the strategic reuse of FOSS in commercial firms. We conducted case studies of 18 commercial firms to understand their usage of open source software. Sixteen of these firms are from non-IT industry segments: e-commerce, retail, manufacturing, banking and financial services, and media; and only two are from the IT segment. We collected data using multiple methods including face-to-face interviews using an unstructured questionnaire, as well as through secondary sources such as company annual reports and articles available in the public domain.

10.3.1 Data Collection

We followed guidelines provided by Eisenhardt [6] for conducting an inductive case study. Our choice of cases was based on an initial screening of companies that are extensive users of open source software. Firms were then shortlisted based on their availability and willingness to participate in the study. The respondents of our interview were either the CIO or a senior level IT manager of the firm. The data collection for these cases was done in late 2010.

While our approach was predominantly a grounded study, we were guided by our literature review and hence started with a base category of variables to structure our interviews [17]. We designed our study as a semi-structured interview based on a questionnaire that covered the following dimensions: quality and productivity gains, scalability and performance, security, flexibility and agility. While the questionnaire was meant to guide the general direction of the interview, the interviews themselves were kept open ended and the respondents were encouraged to digress to any other related topics that s/he found relevant to the subject matter of the study. Given the exploratory nature of the interview, our strategy was to use multiple interviewers. The interviews were conducted by two or more members from the research team and lasted around ninety min. Detailed interview notes including the transcripts of the interview, and the notes containing the researchers impressions were recorded. This was followed by a detailed “within-case” analysis [6] to assess the strategic reuse of FOSS within the firm. Once the relevance of the interview was established, follow-up interviews were conducted, if required, to clarify any open questions that were raised as a result of the within-case analysis. A within-case analysis was performed after every interview and findings from these analyses further guided our subsequent interviews. Thus, the data collection and analysis phases overlapped to a great extent [26]. The output of interviews were captured through interview notes and were then coded [19]. The analysis was conducted within each case separately and then compared across cases to identify cross-case patterns [26].

The initial data analysis followed procedures suggested by King [13] for template analysis. Instead of adopting an open coding approach, we started with an initial set of codes as guided by our literature review and then added/modified them as part of the ongoing data analysis. Detailed case studies were prepared for each of the firms for which data was collected. As part of the case development, excerpts from the interview as well as secondary data were coded using the base category that we started with. New categories were introduced based on the findings from the interviews. We interviewed IT leaders from 18 organizations representing diverse business objectives and found a vast variation in the extent of FOSS usage among these companies. While we provide our findings related to the strategic reuse of FOSS across all these organizations, we include below brief case studies on two of the organizations that considered the usage of FOSS as an essential component of their overall strategy.

A limitation of the approach to use a base category of the variables to guide the interviews is the possibility of not covering any new dimensions that may be of interest. In order to ensure that the interviews cover all possible dimensions, the researchers made specific attempts to probe the managers on other dimensions that might have influenced their choice of FOSS from a strategic perspective. As an example, the inclusion of cost as a strategic advantage (beyond mere economic benefits) was a result of several interviewers citing the strategic benefits derived out of lower cost as further elaborated in Sect. 10.4.

10.3.2 Case Study: Local Search Engine

JustDial operates in the local search engine market in India and is one of the leading players providing services across multiple channels, including print directory (yellow pages), phone/voice search, web search and mobile search. Local search has evolved from word-of-mouth to print directories then to phone-based information services, and on-line web directories. The pre-requisite for an effective local search engine includes the comprehensiveness and the currency of the search index, the relevance of the search result in the context of local information and the ease and speed of accessing information.

The business model for JustDial is based on providing end-customers with people-assisted search data giving information on local Small and Medium Enterprises (SMEs). Customers for JustDial are the end-users who act as the audience and the SMEs who act as the content providers. The revenue stream is through listing fees for sponsored listing; advertisement fees for banners and page impressions; click-based revenue fees; and revenue through database sharing and partnerships.

Local search in India is a fast growing market that is characterized by low capital cost, low switching costs, a large number of competitors, a two-sided market comprising of end users and advertisers, and subject to network effects. Using Porter's five-forces framework, we identify the following strategic imperatives:

- The industry has a relatively low entry barrier for new entrants considering the low capital cost requirement and negligible customer switching costs. To protect against this, JustDial had to invest in developing a very comprehensive local SME database. Ensuring current and relevant information of this database became a strategic necessity.
- With the increasing proliferation of the internet, more and more consumers are opting for web searches instead of using phone-based services. Global search engines such as Google and Yahoo act as default sites and are often used for local information search as well. Competitive needs arising out of this threat include the need for technology-led solutions that can compete against the superior technological powers of the global search engines; and the need to include the smallest of SMEs in their listing that the generic search engines may not be able to include.
- Owing to a nascent market with a high potential for growth, the competition in the field is intense and the basis of competition is not price but providing value addition. Hence, organizations have to constantly offer new and differentiated services addressed through superior technology implementation.

The state-of-the-market observed in this field fits the definition of a hypercompetitive environment with the incumbent players constantly innovating and trying to introduce new and differentiated services. Given such an environment and the constantly changing basis of competition, JustDial adopted a strategy of having their solution offering evolve over time based on market and competitive needs. This imposed the constraint of a very short turnaround time from concept to market. Just-

Dial's choice to compete based on technology meant that the IT systems had to be cutting edge, nimble and flexible. The technical considerations include:

- Need for high reliability and accurate information: constant data collection and integration that needs to be performed with minimal down-time.
- Fast response time (internal target of 95 % of the calls to be addressed in less than 60 seconds). This requires that the Information Retrieval Officers (IROs) comprehend the customer query, identify relevant information, and retrieve and communicate information back to the customer in less than 60 seconds.
- Guaranteed and almost instantaneous SMS (short messaging service) response to user queries requiring superior gateway integration with telecommunication service providers.
- Manage large amount of data: information maintained for four million customers and constantly growing.
- Instantaneous data extraction needing superior search engine capabilities.
- Integration of multiple channels such as phone, web, SMS and WAP.

10.3.3 Case Study: Travel Portal

Cleartrip (www.cleartrip.com) is a leading Online Travel Agency (OTA) in India. The range of products offered by Cleartrip include search and booking reservation for domestic and international flights, hotels, holiday packages, mobile travel services, global destination guides along with "24×7" customer services.

The OTA industry in India is a high growth industry with an increasing number of travelers preferring to use the Internet for planning and booking their business and leisure travel. While the market is concentrated, with the top three (which includes Cleartrip) occupying a bulk of the market share; it is also subject to intense competition and attracts new firms, including large international players. In addition, competition is also provided by airline carriers who operate their own online ticketing websites. The customers of the OTAs comprise the section of society with access to internet and credit cards, and willing to shop online. Given the low switching costs, intense competition and availability of alternative channels; providing a favorable customer experience is of the highest priority for OTAs.

Cleartrip's strategy of providing a superior customer service is captured in their objective of "making travel simple". Their approach to implementing this is through "simple, comprehensive, reliable and responsible" services. Information technology is core to the functioning of an OTA, and plays a strategic role in achieving its objectives. Simplicity is provided through an easy and efficient search mechanism coupled with simple and efficient booking procedures. Cleartrip are market leaders in launching innovative solutions aimed at making online reservations simple, such as the single page view for both onward and return travel; a simple "search, book, go" look-and-feel approach as against a more traditional OTA look adopted by its competitors; and single string search through the "smallworld" application. The Cleartrip platform is integrated with several supplier systems including

airlines, hotels, railways and other travel portals. The smallworld service is offered in conjunction with Yahoo!, Flickr and Lonely Planet. Reliability and responsiveness also requires a high uptime, tight integration with supplier systems, provision of accurate information and reliable payment gateways.

10.4 Data Analysis

In this section, we summarize our findings on the strategic impact of reuse of open source software. The two case studies we presented illustrate the strategic imperatives of IT for these organizations. Here, we outline the advantage derived from FOSS by these two organizations as well as those reported by respondents of other organizations.

10.4.1 Flexibility

Freedom from vendor lock-in and the ability to rapidly customize to changing business needs were identified as the main advantages obtained by open source software in terms of flexibility. The respondents believed that vendor independence was a significant advantage and in more ways than one. Independence from vendor lock-in allowed the firms to upgrade package versions at their own convenience since there was no threat of a version becoming obsolete.

In the case of JustDial, it's strategy of having an evolving system based on dynamic response to market needs meant the need to be free of any sort of vendor lock-in and the ability to rapidly change the IT system. The access to source code rich with contributions from a vast community of developers provided JustDial a constant source of new ideas and an impetus to constantly innovate.

Cleartrip's strategy of using a non-traditional user interface meant that standard applications did not fit their requirement and warranted custom development. The high demand on search efficiency required them to use best-of-breed algorithms. Developing the "smallworld" application required the advanced use of an analytics-based solution. Open source provided them with an adequate repository for identifying and crafting such a solution.

Apart from contractual freedom, vendor independence also meant the freedom to openly innovate and customize the product to their needs, as can be seen from the following quotes:

Impact of vendor lock-in is beyond just contractual or cost implications, it hinders innovation, customization and impedes time-to-market.

The self-service nature of OSS installation packages and independence from vendor personnel for installation speeds up the product installation time significantly

Ability to mix and match various components and the availability of superior building blocks allowed firms to innovate. In addition, visibility to quality source code rich in diverse algorithms and logic also provided firms a stimulus for generating new ideas. Firms were able to deploy more efficient applications that were marked by superior performance, better load balancing abilities and high scalability. Ability to rapidly adapt to evolving customer needs, specifically in web-fronting applications was also a significant source of advantage.

Open Source provides [...] advantage for engineers in that there are readily available designs that engineers don't have to redo. For example, open source frameworks like Symphony, Code Igniter, and Cake PHP give the inherent advantage of quick software development [...] but also enforced pattern programming.

Better functionality, use of FOSS as a starting product base, evolved building blocks and minimized vendor dependency were identified as the prominent factors influencing a shorter development cycle time and a faster time-to-market. Product selection was expedited as there was no need to go through a formal RFP, vendor evaluation and contracting process. Better release management and ease of product installation also provided advantages particularly for IT systems that were subject to frequent changes, such as e-commerce applications.

System development time is reduced by an order of magnitude because of availability of a superior code base that is rich in functionality.

10.4.2 Interoperability

Respondents stated that adoption of FOSS provided them significant advantages with over 90% of the respondents claiming benefits due to the interoperability of FOSS. The study indicates that FOSS adheres to open standards much more than most proprietary options, with FOSS frameworks adopting open architectures. The high interoperability of FOSS products allowed businesses to have hybrid IT installations within their organizations. Additionally, ease of integration of open source products with other applications, including proprietary products, was an important consideration towards adoption of open source.

...its [FOSS] easy installation, simple APIs and good documentation made integration across heterogeneous platforms and frameworks easier.

All proprietary mobile operating systems make it difficult to integrate with their applications. Compare this with Google's open source Android platform which leverages the Web – it opens it up for all other development. The Web has a lot to do with FOSS being successful as well.

Additionally, interoperability allowed businesses to successfully evaluate multiple frameworks in parallel to meet their performance and scalability requirements, encouraging a mix and match of components for a best fit. At JustDial, the business required a tight integration with the SMS gateway to ensure fail-proof delivery of

SMS. JustDial had to ensure that the system was built by carefully evaluating and selecting software components based on specific business and technological needs. Interoperability is also found to reduce vendor dependency and increase the ability to customize and innovate using available tools and frameworks. Interoperability of FOSS components makes it easier to build on top of other components leading to better reuse. As an example, the ability to customize Ubuntu and integrate it with the Mozilla browser to enhance the data extraction performance on the Information Retrieval Officer's workstation was a major factor for its selection in JustDial.

10.4.3 Performance and Scalability

Respondents unanimously agreed that FOSS provided better performance parameters, primarily due to the ability to tweak and fine-tune performance to meet specific requirements. JustDial had stringent performance requirements on their search engine and data extraction algorithms. Open source provided JustDial the ability to choose the hardware platform and develop the software through mixing-and-matching of various open source components that delivered superior performance. One example was the use of Sphinx, a full text search engine that integrates well with the SQL database and provides the ability to achieve fast, relevant and full text search. In addition, the choice of these open source components provided JustDial the ability to scale up and support their aggressive growth targets.

The availability of more than one tried and tested, highly scalable, light-weight, open source framework allows wider choice and makes fine-tuning for performance easier... [In addition] availability of source code gives a sense of ownership vital for sustainable product development.

Scalability concerns are one of the major drivers for FOSS adoption among businesses. FOSS adoption allows businesses to explore multiple options, experiment, customize and innovate without any upfront investments. When finally deployed, it helps scale up rapidly while sustaining the cost advantage. Respondents felt that scalability of FOSS makes it an attractive choice for governments and large businesses as well. With the expectation of high growth, scalability was a key strategic imperative for Cleartrip. Given the need for scalability and load balancing requirements, they chose to deploy a stateless environment. The ability to tune their Apache Tomcat web server enabled them to meet this requirement.

10.4.4 Stability

A majority of the respondents (71 %) indicated that FOSS has been instrumental in creating more stable systems, whereas a small percentage (5 %) felt that FOSS systems are not yet as stable as proprietary software, and the rest indicated no significant differences. FOSS adopters who considered FOSS as stable, observed

that the source code obtained from the OS community is of higher quality and provides a superior platform. These proponents of FOSS were of the opinion that the focus of FOSS has been on intrinsic qualities such as stability and efficiency, and not necessarily just on product features.

This [development] focus is not diluted by any sales pressure. Owing to community involvement, the quality of the product is superior. In addition, the large tester base further ensures low occurrences of bugs and higher stability.

Further, respondents felt that FOSS provided better reliability, as efficiency gains translated to reduced resource requirements, leading to lesser points of failure. The access to source code further aided effective troubleshooting.

Open source helps in troubleshooting and to predict failures, because we have access to source code.

10.4.5 Security

Our study indicates that FOSS provides means to improve security features with 77 % of the respondents claiming security advantages through the reuse of FOSS. Respondents felt that their Linux systems have been significantly more secure than proprietary operating systems. Due to the much larger number of eye balls on the open source code, any security loopholes are spotted quickly and fixed by the community, thus ensuring that FOSS frameworks provide secure components.

Inherent transparency of FOSS wards off unnecessary intrusion. . . There is no motivation to 'hack' open code. . . Often, users find it is easy to debug available source and hence find it more trustworthy leading to inherent security.

10.4.6 Cost Savings

Most firms were of the opinion that while cost savings helped them to justify adoption of open source software, the benefits they accrued from adopting FOSS was much greater than mere cost savings. Firms adopting open source primarily for desktop applications such as Open Office, Thunderbird and anti-virus software that were typically deployed over a large number of desktops reported "cost savings" as their sole driver for adopting open source. Firms adopting open source as their infrastructure layer reported benefits both in terms of cost and superior performance. Further cost savings were reported due to reduced requirement on hardware resources owing to the better performance efficiencies of the open source products. Firms adopting open source for their application layer cited other strategic advantages as the main driver for adopting open source and cost as being an incidental benefit.

Technology-led start-up firms, having a large part of their budgetary expenditure on technology, found the better return on investment (ROI) provided by FOSS as a

strategic lever in attaining early profitability. Cleartrip, being a startup firm, had to build scale rapidly in order to achieve profitability. Technology is a core element in the OTA industry and was a big contributor to the cost structure of the firm. Thus, apart from achieving scale, cost cutting was an important lever to achieve profitability. Achieving high ROI from their technology investment was thus essential. Use of open source software helped them achieve this, resulting in their technology expenses being significantly lower compared to competition. In the case of JustDial, the cost advantage of open source provided JustDial with the ability to do vast in-house development and build a superior technology platform to support the numerous business demands without the worry of astronomical IT development costs.

10.4.7 Types of Reuse

A common type of reuse was mixing and matching of different open source components to achieve flexibility and superior performance. This ability to mix and match coupled with the superior building blocks allowed firms to innovate. Crafting applications by stitching together a diverse set of best-of-breed open source components was a common strategy adopted by most of the respondents.

The ability to modify the FOSS components, not just in the periphery, but also core components that allow the fine tuning of the system functionality for specific performance requirements is another instance of strategic reuse. As an example, JustDial wanted to deploy a performance-intensive search engine. They achieved this by customizing the Ubuntu operating system to tightly integrate with the Mozilla browser and thus obtain superior performance: an ability not easily obtained when developing with proprietary software.

A third kind of reuse was that of knowledge embedded in the open source code. Visibility to the source code, rich in diverse algorithms and logic was reported as a significant advantage from reusing FOSS. Not only were firms able to reuse pieces of quality code from this repository, they were also able to reuse knowledge embedded in this code. Access to specific algorithms and their logic enabled firms to build on them and develop new and innovative ideas.

10.5 Conclusion

We start by answering our first research question on whether firms reuse FOSS to gain strategic advantage. The findings of our survey indicates an affirmative answer. The previous section provides evidence of firms reusing FOSS to gain strategic advantage. For achieving a sustained competitive advantage, firms need to develop superior IT capabilities: specifically, a rent-generating resource that is not easily imitated or substituted [1]. It can be seen that these capabilities consist of not just IT resources but includes a combination of complimentary human and business

resources that provide embedded advantages leading to sustained competitive advantage [23]. In this chapter, we have argued that FOSS can enable firms operating in a hypercompetitive environment with such capabilities providing them with strategic benefits that are essential for sustaining competitive advantages.

In conformance with Sambamurthy [24] that the strategic requirements for firms in a hypercompetitive environment includes agility, continuous innovation and time-to-market considerations, our respondents also rated the flexibility to change and the ability to adapt to changing business needs as a prime motivation for reusing open source software. Flexibility derived through independence from vendor lock-in and the subsequent freedom to innovate were identified as important factors providing firms with the agility to respond to market needs. This is consistent with literature which suggests that in hypercompetitive environments, superior performance is derived from continuously recreating competitive advantage through innovative actions [24].

In addition, better functionality, use of FOSS as a starting product base, evolved building blocks and minimized vendor dependency were identified as the prominent factors influencing a shorter development cycle time and a faster time-to-market. Firms that excelled in the hypercompetitive environment were found to constantly develop their IT capability in iterative loops [25]. The high interoperability of FOSS, the ability to mix and match components and use it in conjunction with existing proprietary software, and the ability to evaluate and adopt FOSS frameworks in gradual phases enabled firms to constantly develop their IT capabilities in tune with business requirements.

The robustness provided by FOSS in terms of stability, security and quality allows these firms to ensure a higher operational efficiency through lesser points of failure, reduced downtime and reduced security breaches. The scalability provided by FOSS allows firms to ensure that they are protected from the need to constantly upgrade their IT systems as they scale up their business. The advantages derived out of lower IT development and operation costs by reusing FOSS were also identified as a strategic benefit, particularly by start-ups with limited resources seeking to maximize the value derived out of their IT investments.

As an answer to the second research question on what attributes of FOSS motivate IT managers to reuse FOSS for strategic benefits, we provide a summary of the attributes and corresponding strategic benefits in Table 10.1.

While we have demonstrated how FOSS can endow firms with these IT capabilities, we would like to stress the possibility of several other strategic advantages that firms can benefit from with the use of FOSS. We attempt to provide two such examples that could perhaps be included as part of future research.

- Sambamurthy et al. [25] posit that IT investments and capabilities influence firm performance through three significant organizational capabilities (agility, digital options, and entrepreneurial alertness) and strategic processes (capability-building, entrepreneurial action, and co-evolutionary adaptation). It is quite intuitive to assume that firms developing IT capability in iterative loops require certain amount of entrepreneurial alertness to constantly search for new FOSS

Capability	Attributes	Strategic benefits
Flexibility	Ability to mix and match diverse set of software components	Agility
	Freedom from vendor lock-in	Faster time-to-market
	Availability of superior code building blocks	Continuous innovation
	Source code as a source of innovation and ideas	Efficient systems design
Interoperability	High interoperability from adherence to open standards and open architecture	Agility
	Enables deploying hybrid installations of diverse components	Faster time-to-market
	Reduces vendor dependency	Efficient systems design
	Allows integrating multi-platform systems	Design efficient systems
Performance	Performance optimization by integrating best of breed components	Agility
Scalability	Open source database found to be highly scalable	Faster time-to-market
	Ability to do modification at the OS level: better integration and higher performance	Efficient systems design
Stability	Better quality providing more stability	Efficient systems design
	Lesser points of failure as a result of reduced resource requirements	Design efficient systems
Security	Reduced security threats leading to higher uptime	Efficient systems design
Cost savings	Enables higher return-on-investment from technology spend	Efficient systems design

Table 10.1: Summary: FOSS reuse for strategic advantage

components that can provide them with new innovations. Studying this link between entrepreneurial alertness and FOSS reuse could be a topic for future research.

- Recent research trends on strategic reuse of IS have focused on the co-creation of IT value across inter-organizational systems [10]. The argument is that the increasing specialization and the faster time-to-market makes it difficult for single firms to assemble the required capabilities to operate in a hypercompetitive

environment, and firms are increasingly collaborating with other firms to co-create IT enabled products and services. It can be assumed that FOSS with its high-interoperability can lend itself well to the co-creation of value across business units and organizations. Studying the role of FOSS in co-creating value could be another area of future research.

As a limitation, we cannot claim the generalizability of our study. Our study provides strong indications of firms gaining strategic advantages through the reuse of FOSS. However, our study does not include firms that have leveraged proprietary software for strategic benefits, nor have we eliminated the possibility of firms having suffered strategic disadvantages through the use of FOSS. The only claim that we would like to put forward is that firms operating in a hypercompetitive environment can potentially gain strategic advantage through the reuse of FOSS. Also, as mentioned earlier, our approach of using a base category of the variables to guide the interviews could possibly lead to some dimensions being missed. While we have made specific attempts to be open-ended and probed for all possible dimensions, we do acknowledge the possibility of missing additional attributes of FOSS that could provide firms with strategic advantage.

It should also be noted that the extent of reuse of FOSS is moderated by the type of license under which the open source software is released. Permissive licenses such as the BSD license allow reuse of code for any purpose whereas restrictive licenses such as the GPL allow reuse only for projects that will be further released under a GPL license [15]. In this chapter, we do not delve on the licensing issues of FOSS, but do want to caution the readers on the implications of the type of open source license on software reuse.

Acknowledgements This research was supported in part by a grant from the Centre for Software and IT Management at IIM Bangalore. The authors acknowledge the assistance provided by Supriya Dey and Uma Bharath in data collection, coding and analysis.

References

- [1] Bharadwaj, A. (2000). A Resource-Based Perspective on Information Technology Capability and Firm Performance: An Empirical Investigation. *MIS Quarterly* 24(1), 169–196.
- [2] Barney, J.B.: *Gaining and Sustaining Competitive Advantage*. Reading, MA: Addison-Wesley (1997)
- [3] Brown, A & Booch, G (2002). Reusing Open-Source Software and Practices: The Impact of Open-Source on Commercial Vendors. In Gacek, C. (Ed.), *Software Reuse: Methods, Techniques, and Tools* (pp. 381–428). Springer Berlin / Heidelberg
- [4] Christensen, C. (1997). *The Innovator's Dilemma: When New Technologies Cause Great Firms to Fail*. Harvard Business School Press.

- [5] D'Aveni, R. A. & Gunther, R. (2007). Hypercompetition: Managing the Dynamics of Strategic Maneuvering. In Boersch, C. & Elschen, R. (Ed.), *Das Summa Summarum des Management* (pp. 83–93). Gabler.
- [6] Eisenhardt, K. (1989). Building Theories from Case Study Research. *Academy of Management Review* 14(4), 532–550.
- [7] Favaro, J., Favaro, K. & Favaro, P. (1998). Value Based Software Reuse Investment. *Annals of Software Eng.* 5, 5–52.
- [8] Frakes W. B. & Kang K. (2005). Software Reuse Research: Status and Future. *IEEE transactions on Software Engineering* 31(7), 529–536.
- [9] Gallardo-Valencia, R. E. & Elliott S. S. (2009). Internet-scale code search. In *Proceedings of the 2009 ICSE Workshop on Search-Driven Development-Users, Infrastructure, Tools and Evaluation* (pp. 49–52). Washington, DC, USA: IEEE Computer Society
- [10] Grover, V., & Kohli, R. (2012). Cocreating IT Value: New Capabilities and Metrics for MultiFirm Environments. *MIS Quarterly* 36(1), 225–232.
- [11] Haefliger, S., von Krogh, G. & Sebastian, S. (2008). Code Reuse in Open Source Software. *Management Science* 54(1), 180–193.
- [12] Henderson, J. C., & Venkatraman (1992). Strategic alignment : a framework for strategic information technology management. In Kochan, T. & Useem. M (Ed.), *Transforming Organizations* (pp. 97–117). Oxford Press.
- [13] King, N. (1998). Template Analysis. In Gillian, S. & Catherine, C. (Ed.), *Qualitative methods and analysis in organizational research: A practical guide* (pp. 118–134). Sage Publications Ltd.
- [14] Krueger, C. W. (1992). Software reuse. *ACM Computing Survey* 24, 131–183.
- [15] Lerner, J. & Tirole, J. (2005). The scope of open source licensing. *Journal of Law, Economics, and Organization* 21, 20–56.
- [16] Madanmohan, T., & De', R. (2004). Open source reuse in commercial firms. *IEEE Software* 21(6), 62–69.
- [17] Maznevski, M., & Chudoba, K. (2000). Bridging space over time: Global Virtual Team Dynamics and Effectiveness. *Organization Science* 11(5), 473–492.
- [18] McFarlan, F. (1984). Information technology changes the way you compete. *Harvard Business Review* 62(3), 98–103.
- [19] Miles, M. B. & Huberman, A. M. (1984). *Qualitative Data Analysis: A Sourcebook of New Methods*. Sage Publications
- [20] Mohagheghi, P. & Conradi, R. (2007). Quality, productivity and economic benefits of software reuse: a review of industrial studies. *Empirical Software Engineering* 12, 471–516.
- [21] Porter M.E.(1980).: Competitive Strategy: Techniques for Analyzing Industries and Competitors. *New York: The Free Press*
- [22] Porter M.E.(1985).: Competitive Advantage. *New York: The Free Press*
- [23] Powell, T.C. & Dent-Micallef, A. (1997). Information Technology as Competitive Advantage: The Role of Human, Business, and Technology Resources. *Strategic Management Journal* 18(5), 375–405.

- [24] Sambamurthy, V. (2000). Business Strategy in Hypercompetitive Environments: Rethinking the Logic of IT Differentiation. In Zmud, R.W. (Ed.), *Framing the Domains of IT Management: Projecting the Future Through the Past* (pp. 245–261). Pinnaflex Educational Resources, Inc.
- [25] Sambamurthy, V., Bharadwaj, A., & Grover, V. (2003). Shaping Agility through Digital Options: Reconceptualizing the Role of Information Technology in Contemporary Firms. *MIS Quarterly* 27(2), 237–263.
- [26] Strauss, A. & Corbin, J. (1990). Basics of Qualitative Research: Grounded Theory Procedures and Techniques. *Sage Publications Ltd.*
- [27] von Krogh, G., Spaeth, S. & Haefliger, S. (2005). Knowledge Reuse in Open Source Software: An Exploratory Study of 15 Open Source Projects. In *38th Hawaii International Conference on System Sciences* (pp. 198b–198b). System Sciences