Nottingham University Business School

UNITED KINGDOM · CHINA · MALAYSIA

Governing Open Source Communities through Boundary Decisions

Wisal Abbas Jaffer Al Bulushi

Thesis submitted to the University of Nottingham for the degree of Doctor of Philosophy

February 2018

Thesis Abstract

Governing open source software (OSS) communities is defined in the relevant literature as the formal and informal means to control and coordinate the collective efforts towards common objectives (Markus, 2007). OSS communities are not based on a fixed structure. Instead, the structure emerges through collaboration. Participants, technical artefacts, ideas, resources, and interactions are fluid (Faraj et al., 2011) in the sense that they are reconfigured over time, depending on the context of the community. This has raised governance challenges in terms of determining "how open is open enough" (West, 2003). Governing a fluid complex technically-mediated ecosystem, such as OSS communities, requires determining whether to keep the boundaries open to all, which may risk the quality of the deliverables, or restricting the contributions to an elite population, which restrains collaboration (Ferraro and O'Mahony, 2012).

In this thesis, I argue that OSS governance is a boundary decision to determine and legitimise the practices that best govern the collective effort in a particular context (Ferraro and O'Mahony, 2012). The current literature focused on two types of boundaries; the external boundary that separates OSS communities from the commercial world; and the role-based boundary that identifies the roles and responsibilities of the individuals (Chen and O'Mahony, 2009). The former boundary has been extensively discussed in the literature by focusing on how firms reap the benefits of OSS products without exploiting the collective effort. The latter boundary focuses on individuals as the main actors of the community.

The current views on OSS governance have two main limitations. First, current accounts focus on creating a governance structure that facilitates the collaboration among dispersed individuals, neglecting the issues of fluidity and dynamicity. As a result, scholars continue to build their studies on taken-for-granted assumptions overlooking the transformations that have occurred to the overall settings of the OSS community. One of the

overlooked areas is the emergence of vertical (i.e. domain-specific) OSS communities, which is the main interest of this thesis.

Second, technology, in the context of OSS, is either considered as an end product or a medium of governance. Current studies failed to address the materiality of technology, where materiality refers to the ways in which the properties of technology are arranged and rearranged in relation to each other to accomplish governance practices in a particular context. The materiality of technology entails different possibilities for governance practices, which is not sufficiently addressed in the literature. Therefore, I argue and demonstrate that any attempt to explain OSS governance without addressing materiality is considered incomplete.

In this thesis, I demonstrate that OSS communities are governed through boundary decisions, where decisions refer to delineating the boundaries of the community. This is achieved by identifying the actors, actions, and resources required to control and coordinate the collaborative effort in a particular context. Boundary decisions entail remaining sensitive to the changes that occur to the context and change the boundaries accordingly.

I adopt grounded theory approach to conduct a case study on Kuali; a vertical OSS community that develops ERP system for the higher education sector. The research findings contribute to the OSS governance literature by developing a theoretical foundation that explains OSS governance as a boundary decision. The emergent theory explains OSS governance in terms of context, control, resources, and materiality. I illustrate through empirical evidence how these constructs interact with each other to govern the collective effort. The thesis contributes to the OSS literature by bringing to the fore the dynamicity and materiality of OSS governance. The thesis also has implications in the area of boundary management. OSS communities represent a non-traditional organisational settings, and thus provides novel theoretical insights with regards to boundary management.

Acknowledgments

First of all, I would like to express my gratitude to my supervisors, George Kuk and Corentin Curchod, for their insightful comments, and continuous guidance and support. I would also like to thank Dr. Creg Shepherd for his support during the early stage of the PhD journey and giving me the confidence to pursue this taxing journey.

Thanks to Kuali community for granting me the access to their archived documents. I appreciate the cooperative attitude of Kuali team members and their willingness to contribute to the research. I am thankful for having the privilege to explore a novel community as Kuali and communicate with interesting and knowledgeable participants.

I extend my deepest appreciation to my husband, Badar, who has been very supportive and caring. Thank you for being there during the most challenging moments of the PhD journey.

To my family members, best friends, and my colleagues at Nottingham University Business School. Thank you for keeping me sane! I appreciate your support throughout my research.

Finally, a special thanks to my children; the joy of my life. Without them, this accomplishment will never be possible.

Table of Contents

List of Figures

List of Tables

Chapter 1: Introduction

1.1. Background of the Study

An open source software (OSS) community consists of diverse and geographically distributed individuals and organisations collaborating to produce software under open source license (Sharma et al., 2002), which allows freely revealing the source code to the public for use, modification, and redistribution (von Krogh et al., 2012). OSS communities represent collaborative communities (Adler and Heckscher, 2006), where the diverse and geographically dispersed contributors pool and govern their shared resources in order to achieve common objectives. The contributors collaborate with respect to agreed-upon rules, values, and norms to produce non-rivalry and non-excludable public good (Franck and Jungwirth, 2003; Dahlander and Magnusson, 2005; von Krogh et al., 2012).

OSS communities have created special forms of organisations to coordinate their efforts and communicate with external entities (O'Mahony and Bechky, 2008). They are recognised as alternative organisational forms that differ from hierarchal and market structures (Adler and Heckscher, 2006; Shah, 2006). OSS communities tend to distinguish themselves from their proprietary counterparts by applying special governance mechanisms to control and coordinate the diverse resources in order to achieve mutual objectives (Markus, 2007). However, the dispersed nature of the community scatters the governance among different entities and artefacts that lack a unified formal organisational structure, and thus create challenges to understand OSS governance (Crowston et al., 2012; Izquierdo and Cabot, 2015).

Governance in general, in both for-profit and not-for-profit organisations, refers to strategic rules and regulations (Markus and Bui, 2012) that monitor, supervise, and control the behaviours of individuals (Lattemann and Stieglitz, 2005; Provan and Kenis, 2008). Governance focuses on all levels of an organisation, ranging from individuals performing mundane

operational tasks to the high level strategic tasks (Markus and Bui, 2012). In addition, governance concerns about motivating divergent individuals to achieve common objectives (Lattemann and Stieglitz, 2005; Schaarschmidt et al., 2015).

In the context of information technology (IT), governance refers to the process of setting a framework for decision rights and accountabilities to manage the formulation and implementation of IT solutions (Weill, 2004). IT governance is an integral part of the organisation's overall governance (Van Grembergen et al., 2004) and cannot be isolated from other assets (Weill, 2004). It ensures the harmony between organisational and IT strategies (Grant and Tan, 2013) by specifying structures, processes, relational mechanisms, and responsibilities that encourage desirable behaviour in the use of IT (Weill, 2004). As the desirable behaviour change, the organisation is required to alter its corresponding governance (Van Grembergen et al., 2004; Weill, 2004). Therefore, IT governance is considered as a dynamic and adaptive process (Weill, 2004; Grant and Tan, 2013). However, with the advancement in the information and communication technologies, new organisational forms have been generated, and therefore require effective IT governance (Tiwana et al., 2013).

OSS communities are considered as an example of these new organisational forms as they have introduced a fundamental new institutional framework to regulate community-driven software development process (Mockus et al., 2002; Watson et al., 2005; Shah, 2006). OSS communities have transformed the software development industry from manufacturer to service provider (Sharma et al., 2002). They became important competitors in the software industry (Lerner and Tirole, 2002; Mockus et al., 2002; Franck and Jungwirth, 2003) and contributed to the development of outstanding software products, such as Linux and Apache (Benkler, 2002; Xu et al., 2005). Therefore, OSS communities need effective governance mechanisms to ensure their sustainability.

OSS governance is described in the literature as the formal and informal means to coordinate and control divergent individuals and organisations to achieve mutual objectives (Markus, 2007). Mainly, OSS governance is discussed in terms of three main perspectives: culture, structure, and process (von Krogh et al., 2012). The cultural perspective describes OSS communities as value driven, and accordingly the community members are controlled and coordinated through OSS ethos (e.g. Stewart and Gosain, 2006), such as altruism, reciprocity, and reputation. The structural perspective highlights the fluidity of OSS communities, which means that the OSS communities are not based on a fixed structure. Therefore, the community cannot be governed using the conventional hierarchal- or market-based governance mechanisms (de Laat, 2007). Accordingly, scholars argue that a governance structure emerges from the communication patterns between the contributors. Hence, different communities form different structures. From a process point of view, OSS governance is explained as a set of rules and practices, such as control, coordination, and communication (e.g. Sharma et al., 2002). However, the existing literature overemphasises on the collaborations among developers at the software development level neglecting the role of the various user groups and the non-technical tasks.

Despite the variations in explaining OSS governance, scholars agree that governance is emergent, adaptive, and evolves overtime. However, the current accounts based their studies on communities that rely on autonomous members, lack formal institutional existence and corporate involvement, and the collaborations are not restricted to a list of deliverables or project deadlines (e.g. Howison and Crowston, 2014; Aaltonen and Lanzara, 2015). This neglects three important points with regards to OSS governance.

First, the existing literature overlooks the transformations that have occurred to the OSS communities since their inception in the mid 1990's. OSS communities have been transformed from autonomous communities

to hybrid forms that merge communities, markets, and hierarchal work practices (Shaikh and Vaast, 2016). In this thesis, I focus on one of the salient transitions that have occurred to OSS communities; the transition from horizontal self-governed OSS communities to vertical restricted communities (Fitzgerald, 2006).

Horizontal refers to the communities that are developing infrastructural applications, such as operating systems and mail servers. These communities, which are dominating the existing literature, are open to wide range of users regardless of their professions and attract developers with generic technical skills. Current studies presume that horizontal OSS communities consist of autonomous individuals, mainly volunteers, who are collaborating without a defined basis of authority. Since these individuals share different interests and motives, OSS governance is mainly described as the means to control and coordinate autonomous individuals in order to develop a reliable OSS product (Markus, 2007).

On the other hand, vertical OSS communities are more focused on producing domain-specific software. Vertical OSS products are co-produced for strategic reasons, targeting specific users, and they are designed by specialised developers as their specifications are more complex and domain-specific. Thus, vertical OSS communities attract individuals that are knowledgeable in that particular domain. Besides, vertical OSS communities are more likely to inherit the work practices of the corresponding domain (or industry) in order to direct the efforts of the community members towards a common objective (Benlian and Hess, 2011). For example, vertical OSS communities tend to rely on employment contracts rather than voluntarily contributions, and the development process is restricted by a predefined list of deliverables. Therefore, governing vertical OSS communities challenges the presumptions that are dominating the relevant literature.

The transformations that have occurred to the settings of the OSS communities lead us to the second neglected aspect of OSS governance, which is fluidity. OSS communities are fluid because they allow resources to freely flow in and out of the community. In fact, the fluidity is considered as the fundamental characteristic of OSS communities as it stimulates collaborations (Faraj et al., 2011). However, fluidity raises challenges with regards to governance because the community contracts and expands unpredictably (Aksulu and Wade, 2010), and accordingly collaborations "*occur in unparalleled scale and scope*" (Faraj et al., 2011:1224). This means that OSS communities are dynamic as they continuously change their boundaries overtime by including and excluding resources based on the context. The dynamic change of boundaries is referred to in the relevant literature as dynamicity (Jarvenpaa and Lang, 2011).

The dynamicity of boundaries in OSS communities is an essential aspect of governance (O'Mahony and Bechky, 2008) because it balances between growth and control (Ferraro and O'Mahony, 2012). This notion of boundaries challenges the conventional perspectives that describe boundaries as static demarcations separating what is inside and outside an OSS community (Jarvenpaa and Lang, 2011). Based on that, I argue that OSS communities are governed through decisions to aggregate and control the fluid resources of the community. These decisions trigger boundary dynamics in the community, and thus they are known as boundary decisions.

Third, the materiality of technology is another neglected aspect of OSS governance, where materiality refers to the ways in which the properties of technology are arranged in relation to each other to accomplish governance practices in a particular context (Barrett et al., 2016). Technology mainly refers to the OSS code, the Internet platform, and the technical artefacts used for development and communication. In the relevant literature, technology is either described as an end product or as a mediator that enables governance practices (Shaikh, 2016). However, in this thesis, I

highlight that technology is inherently dynamic (Tiwana et al., 2010) and, at the same time, inseparable from its surrounding context (Faraj et al., 2011). The properties of technology are not fixed; instead they are arranged and rearranged as the community evolves. The current studies demonstrate that the context of the OSS community shapes how the OSS code is designed. Besides, the design of the code determines the necessary resources and the type of technical artefacts required to facilitate collaborations (Crowston et al., 2012). This means that boundary decisions, i.e. governance, cannot be sufficiently explained without encountering materiality.

This thesis responds to the growing call (e.g. Tiwana et al., 2013) for explaining governance mechanisms that are practiced in dynamic technically-mediated ecosystems, such as OSS communities. I argue that such communities are governed through boundary decisions. Although the importance of boundary decisions was discussed in the early publications related to OSS communities, less attention has been given to their role and consequences with regards to governance.

1.2. Research Questions and Objectives

In this thesis, I aim to answer the following questions: How governance practices emerge in vertical OSS communities? And how OSS community governing contribution shapes its organisational boundary over time? To answer these questions, I have conducted a case study on Kuali; an OSS community that develops and maintains an enterprise (ERP) system for the higher education (HE) sector. Although the primary focus of the thesis is on a vertical OSS community, the research findings contribute to various types of OSS communities, and to a wider range of virtual and online communities.

The topic of OSS governance has been extensively discussed; however the literature failed to offer a definitive definition for OSS governance (Markus, 2007) or a solid theoretical explanation that realises the dynamicity of the

OSS communities and the role of materiality in governance (Crowston et al., 2012). Therefore, I have adopted a grounded theory approach to gain an in-depth understanding of how the community is governed by shaping its boundaries through time. In this thesis, I aim to develop a theoretical foundation to explain OSS governance by highlighting the neglected concepts of dynamicity and materiality.

1.3. The Significance of the Research

The meaning of OSS has changed dramatically during the past two decades (Curto-Millet and Shaikh, 2017). OSS has been transformed from being restricted to back-office infrastructural software products to a software development approach and a strategic business model. Reputable firms adopt OSS as their business models and place their products into OSS communities to attract innovation (Barrett et al., 2013). Linux operating system and Apache web server are the most reputable OSS projects; however there are other prevalent products in various domains. For example, Sendmail email server, BIND domain name software on the Internet, OpenOffice desktop application, Eclipse programming environment, and eGroupware enterprise system (Crowston et al., 2012).

The OSS literature illustrates that OSS is a moving phenomenon (Barrett et al., 2013), which raises governance challenges. Nevertheless, the relevant literature reveals that different OSS communities managed to develop governance mechanisms to aggregate and control the dispersed resources to create value (Hemetsberger and Reinhardt, 2009). This illustrates that OSS communities successfully managed to convert the widely distributed resources into products that compete with commercial software in many different industries (Aaltonen and Lanzara, 2015). Despite their fluidity, OSS communities became pervasive phenomena that have the potential to transform global economy and society (Hallerbach et al., 2013). Therefore, the area of OSS governance offer insights for theory development (Tiwana et al., 2013).

Although the topic of OSS managed to develop a literature of governance of its own (Shaikh and Henfridsson, 2017), scholars failed to develop a unified definition of OSS governance or an OSS governance theory (Crowston et al., 2012). Therefore, in order to explain OSS governance, scholars often borrow concepts from governance theories that were developed for hierarchal- or market-based organisations. These conventional governance mechanisms focus on establishing a structure that supports social coordination and administrative regulation, overlooking the dynamicity of the community (Aaltonen and Lanzara, 2015). Unsurprisingly, the central themes of the existing OSS governance literature is on how to motivate volunteers to contribute to the public good (e.g. von Krogh et al., 2012), and how autonomous contributors collaborate in the absence of a basis of authority (e.g. O'Mahony and Ferraro, 2007). These studies overlook the dynamicity of the community and role of materiality in OSS governance.

The OSS is recognised as competitive software development approach (Benkler, 2002). Besides, it provides potential lessons for other forms of virtual collaboration in software development and in governing distributed resources (Howison and Crowston, 2014). Accordingly, there is a need to understand how such communities govern the collectives while it maintains fluid nature and porous boundaries. Besides, governance routines are embedded in technology. In fact, the collaboration in OSS communities cannot be accomplished without technology. Various technical artefacts are increasingly being integrated into OSS communities. Therefore, explaining OSS governance without including the material aspects of the technology is considered incomplete.

1.4. The Structure of the Thesis

The thesis is organised into eight chapters. Chapter 1 introduces the background of the study, and states the research questions and objectives. It also explains the importance of conducting the research.

Chapter 2 represents the literature review. As the research follows a grounded theory approach, I have started the research by diving into the data with initial insights from the literature. The research involved multiple visits to the literature during data collection and analysis processes. Therefore, chapter 2 is divided into two parts. The first part includes the initial literature review, which represents the current state of the literature with regards to OSS governance. It mainly reviews how scholars define OSS governance. The second part of chapter 2 introduces the concepts that have emerged during the processes of data collection and analysis. It represents the topic of boundary decisions and materiality in relation to OSS governance.

Chapter 3 is concerned with the research methodology. I begin by comparing between the different grounded theory approaches. Then, I clarify the main tenets of the grounded theory approach that was adopted in this thesis. I also explain the rationale behind selecting the Straussian approach for the data analysis. Then, I argue for the suitability of a single case study as the research design. I also explain why the research is targeting the higher education sector. I then provide an overview of the case under study; Kuali, by describing its organisational context. Chapter 3 also explains in details the processes of data collection and analysis.

Chapters 4, 5, and 6 are dedicated to the findings of the study; the story of Kuali. Each chapter refers to a governance phase of Kuali life. Each governance phase is a boundary decision to govern the collective effort and cope with the changes in the context of Kuali.

As the research aims to develop a substantive theory to explain OSS governance, chapter 7 explains the emergence of the substantive theory. I represent the theory through an OSS governance model. Then, I discuss this emergent model in relation to Kuali and also in relation to the existing literature. In chapter 7, I also evaluate the research process using the criteria proposed by Corbin and Strauss (1990).

Finally, I conclude the thesis in chapter 8. I also explain how the emergent OSS governance model contributes to the literatures of OSS governance and boundary management. I also highlight some important practical contributions of the thesis. I then summarise the limitations of the thesis and provide suggestions for future work.

Chapter 2: Literature Review

2.1. Introduction

In line with the grounded theory approach adopted in this research, the literature review was conducted at different stages of the research. Mainly, three major rounds of literature review can be identified; however the review is mainly presented at the beginning of the thesis to set the stage for the subsequent chapters and assist the reader in following the research arguments (Urquhart and Fernandez, 2013; Charmaz, 2014).

The initial literature review was a pre-study to gain an overall understanding on the topic, to clearly identify the research problem, and get familiar with the appropriate methodologies (Glaser and Strauss, 1967). I have focused on how OSS governance is defined and described in the relevant literature. This has provided me with sensitising concepts that acted as the departure point for collecting and analysing the data[1]. The initial literature review allowed me to understand the underlying meaning and concepts in the raw data without forcing any predetermined theoretical frameworks.

The outcome of the initial literature review is illustrated in sections 2.2 and 2.3, where I focus on how OSS governance is explained in the current literature. I argue that due to the fluidity of the OSS community, scholars tend to explain governance by focusing on the daily activities of the community actors (human and non-human). In general, sections 2.2 and 2.3 present the current state of the literature with regards to OSS governance. I highlight that the current studies acknowledge the fluidity of the OSS community, and define governance as emergent and adaptive. Yet, OSS governance is explained using static perspectives.

[1] In chapter 3, I elaborate on the sensitising concepts that have directed the first phase of the data analysis.

The second round of the literature review was performed during data analysis in order to compare the emergent theory with the existing theories. This is necessary to develop a substantive theory (Urquhart and Fernandez, 2013) that advances our knowledge with regards to OSS governance. This included multiple visits to the literature until the research came to an end. The second round of the literature mainly focused on boundary decisions and materiality. These topics have emerged from the iteration between the different stages of the research (i.e. data collection, analysis, literature review). The second round of literature is presented in sections 2.4 and 2.5. In section 2.4, I highlight the limitations of the current literature with regards to the dynamicity of OSS governance practices. I argue that OSS communities are governed through boundary decisions, which is an under-theorised area of research. In section 2.5, I highlight how the literature overlooked the role of materiality in governing the collective effort. Then, I conclude the chapter in section 2.6.

The final round of the literature review was conducted to locate the research findings within the relevant literature. This is represented in chapter 7, where I explain and discuss the emergent substantive theory.

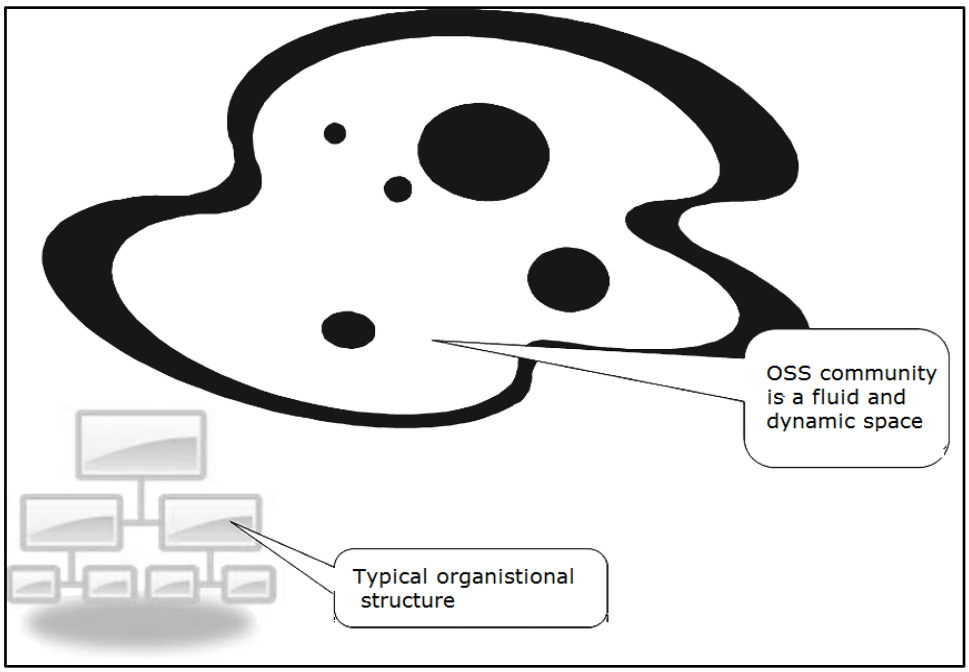
2.2. The Fluidity of the OSS Community

The term open source software (OSS) can be traced back to 1994 (O'Mahony, 2007). It initially referred to software products that are published under open source license, which allows users to access the source code, modify it, and redistribute the modified or unmodified versions (von Krogh et al., 2012). According to West and O'Mahony (2008), openness refers to the transparency and accessibility of the OSS and its associated technical and non-technical processes. Transparency means allowing to view the source code, and its related technical and administrative details, including the discussions and documentations. Accessibility means granting the access to influence the future direction of the OSS product. Therefore, OSS is not the opposite of closed or proprietary

software; rather it refers to the intensity of openness that serves a particular context (Curto-Millet and Shaikh, 2017). The Open Source Initiative; the official body for reviewing and approving OSS licenses, has approved variety of OSS licenses that fulfil the needs of multiple contexts.

The OSS is developed and maintained in an OSS community that consists of a group of heterogeneous dispersed individuals and organisations collaborating through the Internet to produce an OSS product. The community members include, but not limited to, programmers, bug fixers, end users, administrators, foundations, and firms (Dahlander and Magnusson, 2008). They collaborate based on agreed-upon rules, values, and norms (Franck and Jungwirth, 2003; von Krogh et al., 2012). An OSS community is described in the literature in various ways, mainly as an online social production (Benkler and Nissenbaum, 2006; Kane et al., 2014), a technical community (West and O'Mahony, 2008), a computer-mediated interaction (Schaarschmidt et al., 2015), and an internet-mediated collaboration (Hemetsberger and Reinhardt, 2009; Aaltonen and Lanzara, 2015). OSS communities created special forms of organisations (O'Mahony and Bechky, 2008) that have proved their efficiency in converting dispersed knowledge into a pool of shared resources (Aaltonen and Lanzara, 2015). They managed to develop a viable business model that has put proprietary software at risk (Benlian and Hess, 2011), yet not without challenges.

Figure 1: The Fluidity of OSS Community



Adapted from Faraj et al. (2011)

OSS communities are described as fluid and dynamic because the meaning of openness changes overtime within a particular project (Curto-Millet and Shaikh, 2017), and thus they are not based on a fixed structure (see figure 1). It is important to note here that fluidity does not refer to change. Instead, it means that the participants, artefacts, ideas, resources, individuals, and interactions are in flux and unpredictable (Faraj et al., 2011; Shaikh, 2016). This produces highly permeable and dynamic boundaries (Jarvenpaa and Lang, 2011). Besides, OSS communities comprise of geographically dispersed contributors that collaborate through the Internet, and thus resources are fragmented and distributed across dispersed actors (Howison and Crowston, 2014; Aaltonen and Lanzara, 2015). In addition, the collaborative effort in OSS communities mainly evolves around developing a software product that is open, editable, interactive, and distributed (Nyman and Lindman, 2013; Shaikh, 2016). Communicating through the Internet and utilising complex technical artefacts raises the risk of uncertainty and unpredictability in the future direction of the community (Cornford et al., 2010).

The fluidity of the community is considered as an important feature that afford collaboration (Faraj et al., 2011). That is because the absence of a predefined structure gives rise to flexible collaborating mechanisms and tools, and thus allows the flow of resources to enrich the community. On the other hand, fluidity blurs the boundaries of the community, and accordingly raises governance challenges in terms of determining "how open is open enough" (West, 2003). Governing a fluid community requires determining whether to keep the boundaries open to all, which may risk the quality of the deliverables, or restrict the contributions to an elite population, which restrains collaboration (Ferraro and O'Mahony, 2012). This has triggered the need for special governance mechanisms to govern OSS communities (Shaikh and Henfridsson, 2017).

The topic of OSS governance has gained a substantial amount of attention in academia and practice (Aksulu and Wade, 2010). Relevant research spans several fields in management studies and information systems, such as knowledge collaboration (e.g. Faraj et al., 2011), innovation (e.g. von Hippel and von Krogh, 2003), online communities (e.g. Jarvenpaa and Lang, 2011), and virtual communities (e.g. Gallivan, 2001). Therefore, there are widely varying definitions of governing OSS communities that are scattered in the literature. However, OSS governance is best described in terms of observing the day-to-day activities of the community members; i.e. through practice lens. This is further explained in the next section.

2.3. OSS Governance Practices

OSS governance is defined in the relevant literature as the formal and informal means to control and coordinate the collective effort in order to achieve common objectives (Markus, 2007). OSS communities experience a degree of uncertainty and unpredictability due to its fluidity. This encouraged researchers to focus on the daily practices to understand how governance practices emerge and evolve (Chua and Yeow, 2010). While the majority of the literature on OSS governance does not explicitly

reference practice theory, scholars often adopt concepts that are consistent with the practice-based perspective.

Study conducted by	**Objective/Inquiry**	**Data collection method**
Nakakoji et al. (2002)	Investigate OSS collaboration model	-Interview members about how the OSS is developed, what communication tools ae used, how they do business from OSS -Examine mailing lists to understand the communication pattern, which refers to the community structure
Sack et al. (2006)	Understand the role of technology in governance	-Ethnographic observation, text mining of the emails, implementation tool, and documentation to understand how the community structure is formed (roles, decision rights)
O'Mahony and Bechky (2008)	How OSS community manage divergent interests?	-Observation of meeting, technical presentations, conferences -Interviews -Analyse archives: look at detailed interactions, and structural development
Hemetsberger and Reinhardt (2009)	How the community overcomes the challenges of collaboration, and manage divergent interests?	-Observe the communications in Websites, chats, and mailing lists to understand what they are doing -Join the community to understand the "work philosophy"
Chen and O'Mahony (2009)	How an OSS community establishes its boundaries?	-Observe project meetings & conferences -Interviews about membership, sponsorship, decision making -Analyse archives: mailing lists, documentations
Chua and Yeow (2010)	Explore the nature of cross-project coordination	-Analyse the communication and development tools: what they are, how they are used -Trace coordination interactions through observation
Ferraro and O'Mahony (2012)	Understand the membership process	-Analyse membership documents; i.e. who has joined and what role is assigned, the activity of each member -Examine the evolution of the community through interviews and analysing archival documents, such as meeting notes.

Howison and Crowston (2014)	Understand collaboration	-Ethnographic study: participate in an OSS project as a user, subscribe to the mailing lists, follow-up the bug tracking system -Archived-based field study
Li et al. (2016)	How does the fluidity of OSS communities change the stability of governance practices?	-Analyse user activities and their digital traces from activity logs.

Table 1: Examples of Studies that Focused on Daily Practices to Explain OSS Governance

The collective effort of the OSS community is technically mediated. This means that they take place through variety of artefacts and objects. The outcomes of the collective effort is published in the Internet. The governance rules and regulations are implicit and scattered in various technical artefacts, such as the online documentations, email discussions, and bug tracking systems (Izquierdo and Cabot, 2015). Therefore, scholars do not solely rely on interviews, rather they observe and trace the day-to-day activities through logging into the various technical artefact used in the community. In fact some scholars (e.g. Hemetsberger and Reinhardt, 2009; Germonprez et al., 2014; Howison and Crowston, 2014) participated in OSS communities by using the OSS product and joining the communication in order to understand governance. Table 1 provides examples of studies that focused on daily practices to explain OSS governance. The following are the main governance practices that are discussed in the literature.

2.3.1. Social Practices

OSS communities are considered as a type of collective action (von Hippel and von Krogh, 2003), and thus rely on social practices to distinguish themselves from software companies (Ferraro and O'Mahony, 2012). von Krogh et al. (2012) describe the social practices as the "school of virtue" to attract and retain contributions, and to extend the contributions beyond personal preferences. In fact, early publications assert that the absence of social practices demotivates participants (Sharma et al., 2002). The social

practices are explained in the literature in terms of values, norms, and beliefs.

Values, as defined by Stewart and Gosain (2006), are the preferences of behaviours and outcomes of the community participants. A key value that governs OSS communities is reputation (Raymond, 1998; Markus et al., 2000; Dahlander and Magnusson, 2005; Di Tullio and Staples, 2013). Reputation is a valuable asset (Franck and Jungwirth, 2003) that assists in creating an OSS community, motivating participations, and ensuring sustainability (Markus et al., 2000; Di Tullio and Staples, 2013). In addition, contributors are driven by idealistic motives as they assume that their contributions are directed towards a public good (Franck and Jungwirth, 2003). Therefore, OSS communities also promote the values of altruism, reciprocity, knowledge sharing, and gift-giving (Raymond, 1998; Sharma et al., 2002).

Shared norms are the assumptions and expectations within the community. Stewart and Gosain (2006) illustarte how norms govern the collective effort. For example, they argue that the OSS code has to go through formal or informal channels before distribution. Besides, any activity in the community has to be attached to the name of the owner, and the name is never removed without the owner's consent. Another common norm, as explained by Stewart and Gosain (2006), is avoiding forking, where forking refers to the act of splitting the OSS project into independent projects. Although forking does not violate the OSS license, it weakens the community. Moreover, there are agreed-upon norms to incentivise newcomers and familiarise them with the community, such as providing welcoming and guiding documentations (Ren et al., 2007; Hemetsberger and Reinhardt, 2009).

Beliefs refer to the mutual knowledge and understandings of causal relationships (Stewart and Gosain, 2006). The members of the community believe that their work (technical and non-technical) will continuously be

improved by others (Dahlander and Magnusson, 2005), and thus believe that the quality of the code is better than the code in a closed software (Raymond, 1999; Stewart and Gosain, 2006). This suggests that beliefs regulate the relationships among community members, and thus control the behaviour of individuals.

However, with the diffusion of OSS concepts in the commercial world, the aforementioned social practices have been reinterpreted to meet the needs of the business context on one hand, and to ensure the sustainability of the OSS communities on the other hand. Rolandsson et al. (2011) argue that introducing OSS concepts in the software industry reduced the significance of the ideological aspect of the OSS development process. In other words, OSS communities have moved beyond being social movements with limited resources and power (Germonprez et al., 2014). Therefore, the practices that stem from social movements, such as knowledge sharing and gift-giving, are still relevant; however were reconfigured to be business-oriented. In addition, Nyman and Lindman (2013) demonstrate that the shared norms were also redefined to avoid the exploitation of the collective effort. They demonstrate that forking the OSS code became a governance practice to ensure the sustainability of the community. They acknowledge that forking the code may cause effort redundancy; however it ensures the availability of the OSS to a wider range of users. This suggests that the social practices are not fixed. Instead, they evolve depending on the context.

2.3.2. Defining Membership Process

The membership process in OSS communities resolved an essential governance dilemma; how to coordinate the efforts of impermanent members? (Ferraro and O'Mahony, 2012). The membership process identifies who is in and out of the community, and determines the roles of the community members (West and O'Mahony, 2005; O'Mahony and Bechky, 2008). It is also used to regulate the community-firm relationships

by setting clear boundaries between the OSS community and the commercial world (Ferraro and O'Mahony, 2012).

The membership process is apparent since the emergence of OSS communities as it ensures high quality contributions (Markus et al., 2000); however membership criteria differs depending on the context (West and O'Mahony, 2005). In some communities, the membership is open to the public (Demil and Lecocq, 2006), while in others the membership is preserved for certain candidates who meet the objectives of the community (de Laat, 2007; Feller et al., 2008). In some cases, joining a particular community requires testing the technical skills of the members as well as their knowledge regarding the software license to ensure that the members comply with the community rules (Ferraro and O'Mahony, 2012). Moreover, memberships may also require fees to cultivate the community (Rolandsson et al., 2011). In all cases, the membership type selected depending on the objectives and context of the community.

2.3.3. Creating Governance Structure

Creating a community structure is about forming a basis of authority (O'Mahony and Ferraro, 2007), and defining roles and responsibilities (Demil and Lecocq, 2006; Shah, 2006). It is evident that scholars focused on the messiness of the everyday practices of the human and non-human actors in order to understand the community structure (De Paoli and D'Andrea, 2008). Scholars argue that OSS communities do not represent a strict hierarchy nor a completely flat structure (Cornford et al., 2010; Martinez-Torres and Diaz-Fernandez, 2014). Instead, they represent a special governance form (Demil and Lecocq, 2006).

Unlike the hierarchal organisations that rely on authority, and market structures that rely on property rights and price mechanisms, a community structure relies on shared values and norms (Adler and Heckscher, 2006). Therefore, it is essential that OSS community members agree upon a stable structure and pattern of relationships to achieve their mutual goals

(Cornford et al., 2010). The current literature is based on the assumption that OSS communities are not necessarily created with a specific structure in mind. Instead, the structure emerges as a response to the needs of the members and their communication patterns (Nakakoji et al., 2002). Accordingly, the structures that are explained in the literature depend on the case understudy. However, they are mainly described as peer-production community, network, and bazaar.

A peer-production community is a model of social production that is based on volunteerism and self-selection (Benkler and Nissenbaum, 2006). It refers to a group of individuals who collaborate in a decentralised manner to produce information, knowledge, or cultural goods without relying on markets or bureaucratic forms (Benkler and Nissenbaum, 2006). OSS communities are one of the most prevalent form of peer-production communities (Andreev et al., 2010; Morgan et al., 2013) and are characterised by four main features.

First, an OSS community is not associated with a common employer or workplace (O'Mahony and Ferraro, 2007). Second, the community resources are pooled and coordinated to achieve common objectives, which avoids the constraints associated with resource allocation in hierarchies and markets (O'Mahony and Ferraro, 2007; Andreev et al., 2010). Third, community members do not own the output of their effort (O'Mahony and Ferraro, 2007). Fourth, tasks are accomplished and peer-reviewed in a distributed manner to best fit the community objectives (Andreev et al., 2010).

However, scholars interested in this area (e.g. Benkler and Nissenbaum, 2006; Andreev et al., 2010) acknowledged that peer-production settings raise issues with regards to motivation and organisation. Peer-production relies on the efforts of volunteers which cannot be governed neither by central authority nor by monetary compensation. Most importantly, the peer-production perspective does not reflect the current hybrid nature of

OSS communities that merges between OSS and market-driven work practices. Therefore, Feller et al. (2008) demonstrate that peer-production communities are not capable of creating a product that can compete in the commercial world, which is essential for the sustainability of the OSS community. Instead, they suggest that OSS communities resemble a network form.

Networks are suitable governance structures for a project-based community that does not rely on routine-based procedures, and produces an intangible asset (Powell, 1990). Therefore, an OSS community is recognised as a network as it consists of interdependent individuals, firms, and not-for-profit organisations (Feller et al., 2008). These actors collaborate to create a product based on implicit and open-ended contracts that adapt to contingencies (Jones et al., 1997). An effective network provides independent participants the access and the ability to exchange strategic resources (Feller et al., 2008). This raises the challenges of coordinating and securing the exchanges among the community members.

On the other hand, Demil and Lecocq (2006) argue that OSS communities do not resemble a network for two main reasons. First, in a network setting, the identity of the actor and its previous actions are considered as important factors for coordination, which is not required in an OSS community. Second, persistence is an essential attribute of network participants, which contradicts with the OSS concepts. Thus, scholars (e.g. Raymond, 1999; Demil and Lecocq, 2006) suggest that the nature of OSS license promotes a bazaar-like environment.

The term "bazaar" was first coined by Eric Raymond to describe certain stages of implementing an OSS, which differs from the usual cathedral-like approach practised in the commercial world. Raymond (1999) clarifies that particular stages of the OSS development, such as testing and debugging, can be performed in a bazaar-like structure. However, it is impossible to originate and carry out an OSS project using a pure bazaar approach

throughout the development process. Raymond argues that a successful OSS project starts with a well-established design. This indicates that governance structures are not stable, and their dynamicity depends on the context.

However, Demil and Lecocq (2006) used the term "bazaar" not only to reflect the development process, but to include any sort of communication and coordination between OSS contributors. That is because bazaar environment does not impose any sort of restrictions on joining, leaving, and contributing to the community (Rolandsson et al., 2011). Besides, communications with the community members are regulated neither by price nor by formal mechanisms. Although bazaar governance is suitable for information goods, Demil and Lecocq (2006) argue that pure bazaar structure is a failure due to the existence of uncertainty and weak control.

Similarly, Benlian and Hess (2011) argue that, at present, the collective effort of OSS community members is referred to as a "project". This means that the bazaar-like collaboration does not explain the current state of OSS communities. This view has been supported by a more recent research conducted by Germonprez et al. (2014). They argue that the terms "cathedral" and "bazaar" provided rich insights in explaining the governance of OSS communities during the early days. However, these terms are not compelling anymore because nowadays the bazaar has borrowed some of the cathedral's work practices. Software vendors are adopting OSS concepts as part of their business strategy. This indicates that OSS communities are moving towards more formal and structured type of governance.

As a conclusion, each OSS community represents a governance structure that fulfils its specific requirements (Lerner and Tirole, 2002; Di Tullio and Staples, 2013). The practices of the human and non-human actors of the community constitutes the community structure. Therefore, the community structure is not fixed, rather it emerges based on the communication

patterns of the members (Singh and Tan, 2010). Accordingly, the roles and decision-rights are also emergent (Nakakoji et al., 2002; Faraj et al., 2011).

2.3.4. Choosing OSS License

OSS license is considered as the most essential feature that distinguishes OSS from other software products (Demil and Lecocq, 2006; de Laat, 2007). The license ensures that the activities of the members are aligned with the objectives of the community (Bonaccorsi and Rossi, 2003). As OSS communities started to build alliances with firms, the OSS license became an essential governance aspect for defining the identity of the community (Ojha and Rao, 2014). The OSS license has given the OSS the impetus to compete with proprietary software and regulate the commercialisation of the OSS products (Feller et al., 2008; Schaarschmidt et al., 2015).

According to Open Source Initiative (2014), the free availability of an OSS product does not mean that it cannot be used for commercial purposes. The OSS license allows commercialising the software without appropriating it. In other words, the license disallows placing any sort of restrictions to exclude an individual or entity from reaping the benefits of the product. OSS projects gained the interests of for-profit and not-for-profit organisations, and accordingly the Open Source Initiative allowed various types of OSS licenses to emerge in order to serve these interests. Although these licenses differ in the restrictions they apply on the derivative work, they are common in freely revealing the source code of the software to the public.

Licensing agreements inscribe legal and contractual aspects of the OSS community (Hemetsberger and Reinhardt, 2009). The selection of the license is done based on the objectives of the community. Therefore, as the community evolves, the license may change. The current studies demonstrate that the characteristics of OSS licenses have impacts on

attracting and retaining participations, and the overall progress of the development process (Stewart et al., 2006).

2.3.5. Managing Divergent Interests: Resolving Tensions

OSS communities have permeable boundaries that allow resources to freely flow in and out of the community (Hemetsberger and Reinhardt, 2009; Arazy et al., 2016), and thus diverse interests emerge (O'Mahony and Bechky, 2008). Managing divergent interests is a governance practice to reconcile these tensions. In this context, resources include, but not limited to, individual capabilities (e.g. technical and administrative skills), ideas, interests, money, time, and technical artefacts (Feller et al., 2008; O'Mahony and Bechky, 2008; Faraj et al., 2011). In fact, the OSS code is considered as a resource as well (Rolandsson et al., 2011).

The relevant literature is dominated by studies focusing on the individual capabilities as a major resource because they make a difference in practice (Whittington, 2006). Current studies demonstrate that communities attract different capabilities based on the objectives of the community. The firms that spin-out an OSS project to invite innovative ideas attract hobbyists because they are intrinsically motivated, and thus tend to have a higher rate of contribution (Jeppesen and Frederiksen, 2006). On the other hand, domain-specific OSS communities attract knowledgeable participants who are capable of developing functionalities in that particular domain (Fitzgerald, 2006). More recent publications extend the definition of resources form individual capabilities to include any collective arrangements that are directed towards a common goal (e.g. Aaltonen and Lanzara, 2015).

The diversity of resources triggers different sorts of tensions. The classic tension is the one described by Franck and Jungwirth (2003) as the tension between donators and rent-seekers. Donators refer to the individuals and organisations that are contributing to the benefit of the group and not expecting to receive any rewards out of their contributions. In contrary,

rent-seekers are those who act in a self-interest manner aiming for monetary or non-monetary returns from their contributions. Franck and Jungwirth argue that the OSS community needs the effort of both type of contributors, and thus a successful governance model is the one that enables rent-seeking without crowding out donators. Moreover, Kane et al. (2014) highlight a different type of tension; the tension between knowledge retain and change. They argue that the openness of the community continuously invites new perspectives to the community. This requires the community to manage the divergent interests by responding to the new participations while maintaining the current production and knowledge.

The divergent interests and their corresponding tensions are typical in fluid communities, and thus tensions cannot be avoided and they cannot be permanently resolved. In fact, how the community responds to these tensions stimulates governance practices (Faraj et al., 2011). Often, such tensions are resolved by emergent coordination mechanisms without imposing specific rules and structures (Kane et al., 2014; Arazy et al., 2016).

2.3.6. Control

The OSS is freely available over the Internet. However, the community maintains a sort of control to ensure the quality of the software and maintain the integrity of the collective work (Chen and O'Mahony, 2009). Control in OSS communities is either explained in terms of controlling the behaviour of individuals or controlling the code development process. The former is about attempting to ensure that individuals act in a manner that is consistent with the desired objectives (Schaarschmidt et al., 2015). The latter refers to controlling the technical direction of the OSS, such as the design of the software and the schedule of releases (O'Mahony and Bechky, 2008).

The current literature presents different ways to control the behaviours of individuals. The most conventional way is the membership process which

is mainly used to select individuals that are more likely to meet the expectations of the community (Markus et al., 2000). Besides, in the absence of a formal control mechanism, individuals are controlled through social values and beliefs (von Krogh et al., 2012). In addition, OSS communities often practice monitoring and sanction in conjunction with other controlling mechanisms (Sharma et al., 2002). In terms of controlling the code, OSS communities mainly rely technology to facilitate the code development process (Cornford et al., 2010). They also tend to limit the accessibility to certain developers in order to ensure the quality of the deliverables (de Laat, 2007).

2.3.7. Facilitating Collaboration

Facilitating collaboration, i.e. coordination and communication, focuses on the practices of exchanging data (e.g. code, information) between community members, which is the core of the OSS community (Ren et al., 2007). Facilitating collaboration has been considered as a main governance practice since the early publications on OSS governance; however it has been explained from an economic point of view.

Earlier studies in the field explained facilitating collaboration in terms of reducing the transactional cost to produce an economically effective OSS. Those studies assumed that an OSS community lacks defined boundaries, and thus the collaborations are described as unbounded set of agents that utilise unbounded set of resources to accomplish unbounded set of tasks (Benkler, 2002). Therefore, minimising the cost of the transaction is achieved by allowing individuals to self-select the tasks that best suit their capabilities and interests (Benkler and Nissenbaum, 2006).

According to the relevant literature, facilitating collaboration in OSS communities is enabled and constrained by technology (Jarvenpaa and Lang, 2011), where technology in this context refers to the OSS code, and the technical artefacts used for coordinating code development and the communications between the members. Besides, technology also includes

the Internet platform as it promotes the fluidity and dynamicity of the community (Cornford et al., 2010; Ojha and Rao, 2014).

Development tools are used to coordinate and peer-review the development of the source code (Hemetsberger and Reinhardt, 2009). They are used for designing the code architecture, setting standardised coding process, restricting accesses, and scheduling releases (Baldwin and Clark, 2006; Feller et al., 2008). The communication tools include emails, version control systems (VCS), and bug tracking systems. These tools are used to collectively discuss the technical and administrative issues (Hemetsberger and Reinhardt, 2009). They also act as repositories for the communications and discussions to allow the community members to reflect in their work, and look for solutions.

Technology is co-produced by the collective effort and constitute governance practices (Kane et al., 2014). It is apparent that the relevant literature mainly focused on the collaboration between developers at the software project level (Crowston et al., 2007; Chua and Yeow, 2010). Besides, the current accounts are based on the inherent assumption that individuals self-select their tasks (Shaikh and Henfridsson, 2017). This overlooks the emergence of vertical domain-specific OSS communities where roles and tasks are assigned to specific individuals.

In addition, the coordination mechanisms that are discussed assume non face-to-face interactions between contributors, and thus tends to compare the interactions in OSS communities with the conventional face-to-face collaborations (Faraj et al., 2016). This overlooks the fact there is a growing trend of face-to-face meetings between community members, which makes the communication less transparent to the public (Shaikh, 2016). This suggests that the current accounts regarding facilitating collaborations neglects the transformation that has occurred to the overall settings of the OSS communities.

2.3.8. Summary

In this section, I reviewed the main governance practices that are discussed in the literature. It is evident that the practices have been extensively discussed in terms of what actors (individuals and technology) do in order to attract and retain, mainly voluntarily, contributions to develop a high quality public good. It also focuses on what actors do to control and facilitate collaborations.

However, OSS governance practices are not simply about what actors do in order to govern the collective effort. Actually, what actors do is only the point of departure for understanding practices (Nicolini and Monteiro, 2016). Therefore, the study of OSS governance practices requires examining the social and historical conditions of the practices, their scope, the required resources, and their consequences (Hemetsberger and Reinhardt, 2009; Chua and Yeow, 2010). In other words, studying OSS governance practices entails examining all the elements that constitute a practice. Also, examining the traces of the practices and their impact on the wider context.

As I have clarified in the introduction of this chapter (section 2.1), the literature review in the next sections was conducted in parallel with the advanced stages of the data analysis. In section 2.4, I argue that the nature of the OSS allows resources to flow in and out of the community. This means that the boundaries of the community are not fixed, and thus the governance practices are not fixed. Therefore, to better understand governance, it necessary to understand how the boundaries of the community are delineated. I highlight that, although the existing literature acknowledges that the OSS community continuously changes its boundaries, it neglects the role of boundary decisions as governance mechanisms.

During data analysis, it was evident that the materiality of technology is an integral part of governance practices. Materiality refers to how the

community arranges the material properties of technology to accomplish a governance practice. Therefore, in section 2.5, I review the literature on the role of materiality in OSS governance.

2.4. Boundary Decisions

OSS communities are not governed by an individual practice. Instead, context-specific practices govern the collective effort. The governance practices are recurrent within the same OSS community; however they are not identical. "*Each occurrence has a unique combination of actors, artefacts, time, and context*" (Li et al., 2016:3). Therefore, Chen and O'Mahony (2009) demonstrate through empirical evidence that governing OSS communities requires regulating two types of boundaries.

First, identifying the boundary between the community and firms, which regulates the commercialisation of the OSS product and prevents the exploitation of the community efforts. This type of boundary has been extensively discussed in the relevant literature (e.g. West and O'Mahony, 2005; Dahlander and Magnusson, 2008; Morgan et al., 2013), where boundaries are explained as static demarcations that describe what is included in and excluded from the OSS community.

Second, Chen and O'Mahony (2009) also underscore the importance of identifying the boundaries between individuals and organisations. This type of boundary identifies the participants of a particular practice, and their roles and decisions rights. However, this view focuses on creating a governance structure that facilitates the collaboration among dispersed individuals, neglecting the issues of fluidity and dynamicity. In addition, this view highlights the role of human agency in governing the community neglecting the materiality of technology.

A major step in the direction of filling these gaps, scholars (e.g. Jarvenpaa and Lang, 2011; Ferraro and O'Mahony, 2012) directed the attention of OSS governance literature towards boundary decisions. A boundary

decision is situating practices in a particular context by identifying the actors (human and non-human), resources, and domain of action (Ferraro and O'Mahony, 2012). These decisions change the boundaries of the community to cope with changes in the context and objectives of the community.

I begin this section by illustrating that the concept of boundary decision is common in the field of economy; however it had been introduced to the literature of OSS due to the popularity of OSS in the software industry. I illustrate that OSS communities are governed through boundary decisions and I provide examples of boundary decisions from the literature.

2.4.1. Introducing Boundary Decisions to the Literature of OSS

In conventional organisations, boundary decisions are the decisions that delineate the boundaries of the organisation (Barney, 1999). They refer to the activities related to possessing and distributing organisational resources, and defining the domain of action (Ancona and Caldwell, 1990). Boundary decisions were introduced by economists (e.g. Coase, 1937) as a governance mechanism to manage work practices in a particular firm (Barney, 1999). Using concepts from transaction cost economics (TCE), a firm determines the boundaries of its practices by deciding whether to perform the practice within the boundaries of the firm or outsource them using market-oriented governance mechanisms to reduce the cost of exchange and improve efficiency; i.e. build or buy.

Initially, the decision to outsource a practice was highly influenced by the concepts of TCE that focuses on reducing the cost of the transaction. This neglects the importance of the resources that influence the quality of the transaction (Barney, 1999). Performing the practices internally requires specific resources to carry out the activities (Holcomb and Hitt, 2007). Therefore, boundary decisions are considered as strategic complex decisions that do not solely rely on cost-related factors. This has encouraged further studies to understand boundary decisions using agency,

resource-based, and capability-based views, along with TCE concepts (e.g. Poppo and Zenger, 1998; Mayer and Salomon, 2006; Holcomb and Hitt, 2007).

IT outsourcing is a boundary decision to shape the boundaries of IT-related practices in a particular firm (Valorinta, 2011). It was initially explained in the literature from a cost-related aspect. However, the emergence of new technology has dramatically altered the definition of boundaries and boundary decisions (Poppo and Zenger, 1998). Modern organisations are becoming more dispersed in terms of functional, geographical, and hierarchical boundaries. This has called for developing a more comprehensive view of boundary decisions (Santos and Eisenhardt, 2005).

However, it is evident that the current literature on boundary decisions neglects the different types of technology and their impacts on boundary decisions (Lindgren et al., 2008). OSS is an example of a technology that challenges the conventional boundary literature due to the permeability and the dynamicity of the community boundaries that allow resources to freely flow in and out of the community (Faraj et al., 2011). Therefore, scholars argue that the practices handled by OSS communities cannot be considered as outsourcing due to the absence of a centralised organisation and standard software development practices (Winter et al., 2014). Besides, the permeable boundaries hinder the reinforcement of contracts because, according to Dahlander and Magnusson (2008), governing the community members is like herding cats; any attempts to impose control on them scatters them. In addition, OSS communities started as social movements, and thus effectiveness and efficiency were not the ultimate goal of boundary decisions. Instead, fulfilling the intrinsic motives, and enabling knowledge sharing and collaboration were essential aspects that had to be taken into consideration (De Noni et al., 2013).

In the context of OSS governance practices, boundary decisions refer to the delineation of the community boundaries by "*deciding who can*

legitimately participate in, contribute to, and join in an organization's activities" (Ferraro and O'Mahony, 2012:546). In OSS communities, boundary decisions are based on shared understanding and agreement. Therefore, not only they identify the scope and the actors of the governance practices, they also legitimise them (Nicolini, 2013). Boundary decisions became a major concern in the literature of OSS governance due to the growing popularity of OSS development model in the software market. OSS introduced governance practices that involve resources from both the community and the firm. This has caused blurred boundaries around the community, and thus challenged governance practices (Jarvenpaa and Lang, 2011).

Boundary decisions have been discussed in the literature of OSS governance; however implicitly. In the rest of this section, I argue that boundary decisions shape the governance practices, and shape the community boundaries. Unlike the existing literature that describe boundaries as static demarcation that separates community from firm, and as segregation between actors and resources (e.g. Benkler, 2002), I illustrate that boundaries also integrate the dispersed resources and the collective effort. The following are examples of boundary decisions that were evident in the relevant literature; however implicitly considered as governance mechanisms.

2.4.2. Examples of Boundary Decisions

Fitzgerald (2006) argue that the OSS communities have been incrementally transformed to new generation OSS that balances between market-driven and OSS values. These transformations are not only triggered by the size of the community and the line of codes (de Laat, 2007; De Noni et al., 2013). Instead, they emerge from ongoing boundary decisions to cope with dynamic changes in the requirements of the OSS community (Kane et al., 2014).

In the following subsections, I explain how OSS communities have changed since their inception in the mid 1990's. Even though they have not been labelled as such, the following are examples of boundary decisions. They illustrate how OSS communities are governed by including and excluding actors and resources based on the changing context of the community. This means that boundary decisions determine the practices that achieve the objectives of the community in that particular context.

2.4.2.1. Autonomous OSS Communities

The early explanations of OSS governance stemmed from the ideological direction of the Free Software Foundation (FSF). Originally, the FSF was found in the 1980's to create communities that share and develop software beyond intellectual property rights (von Krogh et al., 2012). The FSF considered any sort of restrictions on the transparency and accessibility as immoral (von Krogh et al., 2012). Therefore, the proponents of FSF believe that the source code has to be freely available to all for use and modification. In addition, they enforce derivative work to be distributed as "free software" and all changes to be returned to the original author. This was essential to avoid firms from exploiting and commercialising the collective effort of FSF (West, 2003).

In the mid 1990's, some members of FSF started to take a more pragmatic direction by negotiating the degree of openness with the software industry in order to enhance the quality of the produced software (O'Mahony, 2007). These members initiated the OSS movement. They believed that the source code has to be open rather than free. The OSS proponents were moderate in comparison to the radical FSF camp (de Laat, 2007) as the former allowed individuals and firms to customise and redistribute the co-produced software as they desire (West, 2003). Overtime, the OSS movement evolved and moved apart from FSF (Raymond, 1999).

Despite the differences between "open source" and "free software" camps, early publications in the field of governing OSS communities were

influenced by the tenets of FSF and considered openness as the opposite of closed software (e.g. Ljungberg, 2000). Therefore, OSS communities were described as autonomous because they are initiated by individuals or groups and grow organically overtime; independent from any organisation (de Laat, 2007; West and O'Mahony, 2008). de Laat (2007) describes the governance practices in such communities as spontaneous because such communities include group of hackers who are challenging the software industry by defending the right of publicly distributing the source code of the software products (de Laat, 2007; O'Mahony and Bechky, 2008). The members spontaneously govern the collective effort without any explicit coordination and control mechanisms. Adopting spontaneous governance practices is compatible with the context of a community that is dominated by volunteers. The governance practices are mainly directed towards the daily activities of the developers in the project level of the community (Raymond, 1999).

Boundary decisions in autonomous communities, as summarised in table 2, mainly focus on legitimising OSS governance practices that are capable of maintaining a competitive OSS (Benlian and Hess, 2011). Scholars consider social practices as a precondition for any formal and informal governance mechanism (Markus et al., 2000). Practices that assure trust, reputation, and altruism (Stewart and Gosain, 2006) are necessary to motivate volunteers in co-producing high quality software (Lakhani and Wolf, 2005). Besides, the community highly relies on voluntarily contributions. Therefore, contributors join the community by their own will and participate according to their objectives and interests (Kane et al., 2014). Accordingly, memberships are fluid (Sharma et al., 2002) causing porous boundaries that allow greater inflow of resources (Faraj et al., 2011; Aaltonen and Lanzara, 2015). Members self-select their tasks, where tasks are not associated with any particular attributes, such as qualification (Nakakoji et al., 2002). In addition, the community does not adhere to project deadlines or list of deliverables (Mockus et al., 2002; Sack et al., 2006).

Boundary decisions: based on emergence (i.e. spontaneous)
Social practices: -Rely on values and norms to attract volunteers
Roles and responsibilities: -Participants self-select their tasks -Roles are not associated with attributes (e.g. qualification) -No project deadlines -No list of deliverables
OSS License: -Non-restrictive license -Mainly non-excludable, non-rival
Membership: -Participants join according to their own will -Fluid: resources and participants freely flow in and out
Control: -Meritocracy -Pluralistic -Clan control

Table 2: Boundary Decisions in Autonomous OSS Communities

The fluidity of membership has encouraged exercising control based on meritocracy. This means that individuals with higher technical and administrative skills are more likely to hold authoritative positions, and thus control the future direction of the project. Control in autonomous communities is pluralistic (O'Mahony, 2007). Scholars argue that the conventional control mechanisms, such as output control and behavioural control, cannot be applied in autonomous OSS communities due to the lack of a centralised organisation (Schaarschmidt et al., 2015). The informal clan control is the most evident control mechanism in autonomous communities (von Krogh et al., 2012). Clan control focuses on minimising divergent interests to emerge through socialisation (Schaarschmidt et al., 2015).

2.4.2.2. Sponsored OSS Communities

As the OSS approach gained momentum in the software industry, the meaning of openness was altered to refer to the intensity of code transparency and the degree of accessibility to the development process (West and O'Mahony, 2008). Autonomous communities have realised the need to be sponsored and move beyond the efforts of individuals. This was achieved by adding sort of formality to their structure and processes (O'Mahony and Bechky, 2008). Therefore, the perception of openness in the relevant literature has moved beyond the binary conception of open versus closed software to OSS as a business strategy to reap the benefits of openness (Dahlander and Magnusson, 2005). Besides, sponsorship has emerged as an important feature of an OSS community (O'Mahony, 2005). In other words, OSS communities are more likely to be sponsored rather than autonomous.

Sponsored OSS communities take various forms. They may refer to a group of individuals working on an OSS project that was internally developed by a firm without a prior community (Schaarschmidt et al., 2015). Then, the firm releases the code to the public in order to support, diffuse, and extend the project (West and O'Mahony, 2005). A sponsored OSS community also refers to an autonomous community that created a legal shell (de Laat, 2007) to protect its rights with the popularity of OSS in the business world, and to plan the future direction of the community (O'Mahony, 2005; West and O'Mahony, 2005). The community sponsor, which could be a firm, a government agency, or a not-for-profit organisation (West and O'Mahony, 2005), is the governing body of the community and controls the code development process (West and O'Mahony, 2008).

Boundary decisions: balance between growth and control
Roles and responsibilities: -Roles with titles and attributes
Membership: -Consistent/stable membership
Control: -Homogeneous -Restricting transparency and accessibility

Table 3: Boundary Decisions in Sponsored OSS Communities

One of the main objectives of the sponsorship is to attract contributions in order to allow the community to grow and thrive. At the same time, the sponsor has to control the contributions to assure efficiency and quality (West and O'Mahony, 2008). Therefore, sponsoring an OSS community is a boundary decision that focuses on balancing between growth and control. Sponsoring a community increased the sponsor's desire to control the development process in order to ensure quality, security, and sustainability. Accordingly, control in sponsored OSS communities is homogenous, where the sponsoring body is the centre of control.

Unlike autonomous communities that rely on spontaneous governance practices, sponsored communities have their internal explicit and formal tools for governing the collectives, such as code modularisation, division of roles, and a clear decision-making process (de Laat, 2007). As summarised in table 3, sponsored OSS communities identify roles that are associated with clear descriptions and attributes (de Laat, 2007). In addition, the membership process is more stable in comparison to autonomous communities (O'Mahony and Bechky, 2008).

2.4.2.3. Hybrid OSS Communities

OSS communities have reached to a level of success and maturity such that they cannot ignore the outside world (de Laat, 2007). Therefore, some OSS communities tend to form a hybrid community that includes a combination

of market, community, and hierarchy in order to fulfil the heterogeneous requirements of the members (Demil and Lecocq, 2006; Shah, 2006; O'Mahony and Ferraro, 2007). A hybrid OSS community builds alliances with diverse service providers to offer payable complementary services, such as consultation, data migration, support, and training (Bonaccorsi and Rossi, 2003). These service providers also contribute to the development of the resources and expertise within the community, and accordingly improve the quality of the OSS product (Stewart et al., 2006). It is important to note here that the hybridity is not limited to the community structure. Hybridity may also include designing the software in a way that merges OSS and proprietary software development processes (Sack et al., 2006), or mixing face-to-face and virtual communication patterns between the community members (Watson-Manheim et al., 2012; Kane et al., 2014). In general, forming a hybrid OSS community is a boundary decision to fulfil the heterogeneous requirements of the community.

Scholars demonstrate through empirical evidences that the OSS communities tend to adopt a hybrid approach to handle the shortcomings of the OSS communities and harness the benefits of both traditional and OSS software development models (Sharma et al., 2002). There is a consensus in the literature that OSS development has various advantages over the proprietary software development approach (Deodhar et al., 2012). For example, OSS development process minimises time-to-market, and involves lower software and hardware costs (Sharma et al., 2002; Singh and Tan, 2010). Therefore, software companies adopt OSS development process as part of their business model.

Moreover, other scholars argue that the software industry form a hybrid OSS community to resolve innovation issues (Jeppesen and Frederiksen, 2006). Bonaccorsi and Rossi (2003) argue that software companies release one or more of their products to the public in order to diffuse the product, which is an important stage of innovation process. This has allowed the proliferation of different OSS license agreements that balance between the

market needs and the OSS concepts. The hybrid model facilitates the innovation process by securing complementary services to the community members and providing incentives to the development of the "mundane but necessary tasks", as described by Bonaccorsi and Rossi (2003). The high reliance on the Internet and the availability of various technical artefacts assists software companies to target and attract the suitable external knowledge (Liang et al., 2016).

The hybridity of the OSS communities raises enquiries with regards to the governance practices that balances the community-firm relationships (De Noni et al., 2013). Initially, conventional OSS communities drew a rigid boundary between the industrial and OSS work practices. However, the transition to a hybrid community requires both the OSS community and the software industry to reconfigure their work practices, and thus negotiate their boundaries. Therefore, governance practices in hybrid OSS communities aim to merge the benefits of OSS and the business world, as summarised in table 4.

Boundary decisions: fulfil heterogeneous requirements
Roles and responsibilities: -Attract capabilities that meet the objectives of the community -Invite diverse resources
Membership: -Firm impose their strategic rules -Joining the community according to certain criteria
OSS license: -Restrictive OSS license
Control: -Tight control -Hierarchal control
Facilitating collaboration: -Securing communication -Standardising the technical artefacts

Table 4: Boundary Decisions in Hybrid OSS Communities

The boundary decisions in hybrid communities are explained in a study conducted by Shah (2006). The study compares between the governance of a "gated" and an "open" OSS communities; hybrid and autonomous respectively. Shah demonstrates that each community experiences different boundary decisions. In terms of attracting participants, it is evident that the "open" community relies on social practices. In contrary, the "gated" community imposes a restrictive OSS license and sets clear rules to align the actions of individuals with the overall objectives of the community. Besides, the "gated" community exercises tighter control on the code development process. In terms of resources, Shah demonstrates that the "gated" community attracts the interests of professional developers, while the "open" community attracts hobbyists, who are often driven by fun and enjoyment.

Moreover, Rolandsson et al. (2011) studied the consequences of adopting OSS concepts as a part of a company's strategy. This study demonstrates how an OSS community reacts to the tensions caused by merging market strategies with OSS approach. Firms have to impose rules that align the OSS practices with the strategy of the firm. This has triggered the need to build a hierarchal control in order to balance between openness and bureaucracy. Merging industrial and OSS practices also raises the need to secure the communications among the community members.

In summary, creating a hybrid OSS community invites diverse resources that are likely to directly or indirectly influence the governance practices of the community (Schaarschmidt et al., 2015). This means defining a restrictive membership and granting the access only to those who meet certain criteria (Hemetsberger and Reinhardt, 2009). The more the community is dependent on external resources, the more it restricts the membership process (Chen and O'Mahony, 2009). Moreover, the literature on motivation has been redirected from motivating individuals to motivating firms to engage in OSS projects (O'Mahony, 2007).

2.4.2.4. Vertical OSS Communities

A substantial amount of the literature focuses on how contributors are motivated to build the community, share a basis of authority, and govern themselves in a community that creates and maintains a horizontal OSS product (Fitzgerald, 2006). As mentioned earlier, horizontal refers to infrastructural applications- such as operating systems, web servers, and databases- that are characterised by standard technical requirements. It is evident that OSS products are increasingly moving from back office invisible infrastructures to domain-specific vertical domains that serve particular industries (O'Mahony, 2007; Benlian and Hess, 2011). Vertical OSS communities introduced OSS concepts to professional software development processes in conventional organisational settings. They are targeting specific industries and domains, and thus context became an important aspect in governing an OSS community (Shaikh, 2016).

In horizontal communities, the participant's willingness to contribute is the essence of the development process (Howison and Crowston, 2014). Therefore, self-governed participants communicate via the Internet in the absence of contracts and managerial authority (Aaltonen and Lanzara, 2015), and borrow elements of organisation only as required (Winter et al., 2014). In contrary, vertical OSS communities are domain-specific (Fitzgerald, 2006) and consist of knowledgeable individuals collaborating according to work practices that are relevant to that particular domain (Rolandsson et al., 2011). Therefore, boundary decisions in vertical OSS communities ensure that the governance practices are domain-specific; i.e. aligned with the objectives of the domain.

Unlike horizontal OSS communities that attract wide range of contributors, vertical communities rely on the efforts of selective knowledgeable group of developers and users to maintain the mission of a particular domain. Therefore, the membership process is not open to the public; rather it is selective and based on certain criteria (Ferraro and O'Mahony, 2012).

Accordingly, the produced software is published under a more restrictive OSS license (de Laat, 2007).

Boundary decisions: create domain-specific community
Roles & responsibilities: -Select domain-specific resources -Selective knowledgeable individuals
Membership: -The domain/industry impose its strategic rules -Joining the community according to certain criteria
OSS license: -Restrictive OSS license
Control: -Gated OSS development process
Facilitating collaboration: -Securing communication -Standardising the artefact

Table 5: Boundary Decisions in Vertical OSS Communities

According to Fitzgerald (2006), one of the main differences between horizontal and vertical OSS communities is the software development process. In a vertical community, the development process is gated. In other words, although the source code and the community collaboration is transparent; the accessibility is limited to a selective group of individuals. In addition, horizontal and vertical communities vary in their development steps. A typical software development process includes planning, analysis, design, and implementation. Fitzgerald argues that in a horizontal community these steps are concatenated and performed by the same developer because the requirements of horizontal applications are generally understood. On the other hand, the development process in a vertical community is performed by multiple individuals. The planning is often assigned to the key players of the community to maintain domain-specific objectives. Requirements analysis and design are more complex and

performed by knowledgeable individuals. Besides, the requirements of a vertical OSS are explicit and well documented.

Crowston et al. (2012) support the findings of Fitzgerald and explain that the software development steps are neglected in horizontal communities because the project team has a common understanding of the system. Besides, the requirements are scattered in email discussions and bug reports. This means that the governance practices of vertical OSS communities challenge the taken-for-granted assumptions in the relevant literature, and thus require further studies (Aksulu and Wade, 2010; Benlian and Hess, 2011).

2.4.3. Summary

The aforementioned examples illustrate how OSS communities evolved from an ideology to a business model (Deodhar et al., 2012). It is important to point out here that these examples should not be considered as chronological changes that has occurred to the OSS community. Instead, these are illustrations to show that OSS communities continuously change their forms. They may change from one form to another, and also may combine the features of multiple forms based on the overall context (O'Mahony and Ferraro, 2007).

An OSS community begins with a certain form and gradually attracts diverse stakeholders to ensure the survival of the community. However, the newly joined stakeholders have their own objectives and interests that may not totally overlap with the direction of the community. Similarly, with the advancements in information and communication technology, OSS communities also include and exclude technologies based on their requirements. Accordingly, the community continuously change the boundaries to govern the collective effort.

The current literature illustrates that OSS communities combine both dynamic and stable elements. On one hand, OSS communities are fluid and

comprise of porous boundaries that encourage dynamicity. On the other hand, the community maintains a certain form of stability in order to be governed (Arazy et al., 2016). This fluidity is not only a characteristic of OSS communities; rather it's essential to make governance possible (Faraj et al., 2011). However, current accounts on governing OSS communities focused on creating a governance structure that facilitates the collaboration among dispersed individuals, neglecting the issues of fluidity and dynamicity (Aaltonen and Lanzara, 2015). Future research should further our understanding of how boundaries emerge, maintain, change over time, and what triggers these changes (Crowston et al., 2012).

It is also evident that these examples mainly focused on delineating boundaries around individuals as the key participants, neglecting the vital role of technology in governance practices. In a recent literature review, Crowston et al. (2012) urged scholars to attend to the materiality of technology in governing the community. For example, they argue that the type of technical artefact in-use is an important input. Besides, the interactions between various technical artefacts have implications on governance. This highlights the importance of materiality in governing OSS communities, as will be further clarified in the next section.

2.5. The Role of Materiality in OSS Governance

The aim of this thesis is to develop a conceptual foundation to theorise OSS governance. However, this cannot be achieved without taking into consideration the materiality of the technology for its powerful implications on governance (Shaikh and Henfridsson, 2017). As mentioned previously, materiality, in the context of this thesis, refers to the ways in which the properties of technology are arranged and rearranged in relation to each other in order to entail different possibilities for governance practices in a particular context (Barrett et al., 2016).

Therefore, to understand governance, it is essential to understand how technology is designed, used, and redesigned to accomplish governance

practices. In the relevant literature, technology is expressed in a variety of ways. Throughout this thesis, the term "technology" refers to the OSS code, and the various technical artefacts used for communication and coordination, such as wikis, emails, code version control, and bug tracking systems. It also refers to the technical platform necessary for the community to function and survive.

As argued in the previous section, OSS communities are governed through decisions that trigger boundary dynamics, i.e. boundary decisions. There are various governance practices that are evident in OSS communities. Boundary decisions determine the practices that fulfil the requirements of the context, and identify the required resources, where technology is considered as an essential resource. However, the OSS code and the technical artefacts are editable and configurable (Cornford et al., 2010; Nyman and Lindman, 2013) in order to accommodate the changing rules and principles of the surrounding context (Hemetsberger and Reinhardt, 2009). This means that both the OSS community and technology are in flux.

The existing literature mainly focuses on how technology enables and constrains governance mechanisms in OSS communities. For example, scholars extensively explain how email and forum discussions enable the categorisation of the members in terms of their interests and expertise, which allows task distribution and coordination (e.g. Barcellini et al., 2008; Bird et al., 2008). In addition, there are various studies that focus on how the development artefacts, such as the version control system, are capable of drawing the dispersed community together and coordinating the software development process (e.g. Cornford et al., 2010; Shaikh and Henfridsson, 2017). However, what has been overlooked is the fact that the properties of technology are not predetermined and not arranged in isolation. Instead, these properties become meaningful and consequential only when they are used in-practice. In other words, the role of materiality

in governing OSS communities is an overlooked area of research (Tiwana et al., 2013).

According to Kallinikos et al. (2013), OSS communities represent complex ecosystems where the design and use of technology are in flux. Hence, technology is in a constant state of change; incomplete. This incompleteness, according to Kallinikos et al. (2013), represents both an opportunity and a problem. It is an opportunity because it accommodates the fluidity and dynamicity of the surrounding context; however it is a problem because it raises challenges with regards to the stability and control of the design and use of the technical artefact. Based on that, in this thesis, I focus on the materiality, i.e. emergent properties of technology, rather than focusing on technology per se.

Materiality is a relatively new concept in the context of online communities, and OSS communities in specific. However, scholars (e.g. Orlikowski and Scott, 2008; Leonardi, 2011) have called for attending to the role of materiality in the change and stability of organisations. They argue that, with the advancements in information and communication technologies, it is essential to adopt an inseparable ontology when theorising the relation between objects, entities, actors, and work practices. Therefore, in this thesis, I adopt a relational perspective in theorising materiality. I argue that the material properties of the technology are defined and redefined through interactions, and influence the dynamicity and stability of the OSS communities, i.e. influence boundary decisions.

There are recent modest attempts in the literature that highlight how the objectives of the community are entangled with the materiality of technology (e.g. Aaltonen and Lanzara, 2015; Shaikh, 2016). These studies demonstrate that the properties of technology are not fixed. Instead, they continue to evolve even after implementation, and thus generate new forms of agency that cope with the new objectives and settings of the community (Bolici et al., 2016; Shaikh and Henfridsson, 2017). For example, Shaikh

and Vaast (2016) conducted a study on the communications available in a version control system (VCS) in Linux community, where communications were initially open and transparent. They studied a decade's worth of communications and demonstrated that as VCS is used overtime, it has created temporary "pockets" that are less transparent. They argued that these spaces address issues that cannot be solved by the community in the conventional open spaces. This suggests that the materiality of technology is relational in a sense that the properties of VCS evolve and emerge in relation to the surrounding context. This is considered as one of the main points of departure for developing a theoretical foundation to explain OSS governance.

2.6. Conclusion

The literature review stresses two main shortcomings. First, the role of boundaries in governing OSS communities is under-theorised and lacks empirical evidence (Faraj et al., 2016). There are few attempts to explain governance practices through boundary decisions in OSS communities because it is difficult to capture the stability of the practices while considering the fluidity using the conventional structural-based perspectives (Li et al., 2016). The current accounts focus on creating a governance structure that facilitates the collaboration among dispersed individuals, neglecting the issues of fluidity and dynamicity. As a result, scholars continue to build their studies on taken-for-granted assumptions overlooking the emerging issues that has occurred to the overall settings of the OSS community.

Therefore, there is a mismatch between the previous studies and the emerging issues regarding OSS communities, which calls for a conceptual revolution (Winter et al., 2014). One of the neglected areas of research is how vertical OSS communities are governed. In recent years, it is evident that OSS communities are moving towards vertical domains; i.e. becoming domain-specific, as explained in section 2.4.2.4. These communities

challenge the presumptions that have dominated the relevant literature. For example, joining vertical communities is restricted to knowledgeable (i.e. specialised) contributors, which differs from the volunteer-based contributions discussed in the literature. Besides, OSS governance practices are mainly explained in terms of communities with high degree of uncertainty and unpredictability. However, vertical communities are domain-specific and are created for a specific purpose, and thus less ambiguous. These different settings trigger the need for further studies.

Second, technology is considered as either an end product or a medium of governance (Shaikh, 2016). Current studies failed to address the role of the material characteristics of the technology in accomplishing governance practices (Faraj et al. 2011). How the properties of the OSS code and the technical artefacts are arranged and re-arranged in relation to each other to govern the dynamic community remains under-theorised. In this respect, the research questions are: How OSS governance practices emerge in vertical OSS communities? How governance practices shape the boundaries of the community overtime? Table 6 lists the definitions of key terms used throughout the thesis.

Term	**Definition**
Governance	Formal and informal means to control and coordinate the collaborative effort to achieve mutual objectives. Example of governance practices: control, managing divergent interests, defining membership (see section 2.3)
Boundary decisions	Identifying and legitimising the governance practices, their actors, and required resources based on the context. These decisions trigger boundary dynamics in the community.
Technology	Refers to the OSS code, and the various technical artefacts used for communication and coordination, and any technical platform/software necessary for the community to function and survive.
Materiality	The ways in which the properties of technology are arranged and rearranged to entail different possibilities for governance practices in a particular context

Table 6: Main Definitions used in the Thesis

Chapter 3: Methodology

3.1. Introduction

In chapter 2, I have summarised the literature on governing OSS communities. I have illustrated that the current accounts explain the dynamicity of the governance practices using fixed structural mechanisms. Besides, the current literature neglects the role of materiality in OSS governance practices. In this chapter, I describe how I have designed the research framework in relation to my research questions: How OSS governance practices emerge in vertical OSS communities? How governance practices shape the boundaries of the community overtime?

I begin this chapter, in section 3.2, by clarifying the grounded theory approach adopted in this research. Then, in section 3.3, I explain the rationale behind selecting a single in-depth case study as the research design. Then, I describe the organisational context of Kuali; the case under study. Although the processes of data collection and analysis were conducted simultaneously in an iterative manner, I explain them separately and sequentially for clarification purposes in sections 3.4 and 3.5 respectively. Finally, I summarise the chapter in section 3.6.

3.2. Research Framework: Grounded Theory Approach

The ultimate purpose of this study is to build a substantive theory that explains OSS governance. The focus of the research is on vertical OSS communities that challenge the taken-for-granted assumptions in the relevant literature. As explained in chapter 2, there is a lack of theoretical foundation to assist researchers in explaining the emergence and dynamicity of the OSS governance practices, especially in vertical OSS communities. Therefore, in this thesis, I follow a grounded theory approach that is suitable for phenomena that are emergent and poorly understood (Corbin and Strauss, 1990).

Grounded theory is an inductive data-driven methodology (Walsh et al., 2015). It aims to develop theory that is grounded in empirical data, rather than setting hypotheses and testing theories. Accordingly, I have started the research without a definitive theoretical basis. The research was driven by sensitising concepts from the initial literature review. The aim was to develop empirically-driven theoretical explanation with regards to OSS governance.

It is evident that there is an increase interest in grounded theory approach in the field of information systems over the past decade (Urquhart and Fernandez, 2013). However, grounded theory was criticised for producing low level of theory development due to the misuse of the grounded theory concepts (Urquhart et al., 2010). Therefore, I begin this section by providing a brief historical background about grounded theory and the emergence of the contentions and conflicts in the adoption of grounded theory. Then, I explain the grounded theory approach that is adopted in this thesis.

3.2.1. Types of Grounded Theory

Grounded theory was initially developed by Barney Glaser and Anselm Strauss, which was represented in their famous book "The Discovery of Grounded Theory" in 1976 (Corbin and Strauss, 2008). The aim of the book was to provide an alternative to the deductive approaches that require precise hypotheses to be developed before data collection (Kelle, 2010). Emergence is the key principle of grounded theory, where the research process and outcome emerge from data, rather than forcing categories on data. Besides, the researcher should employ theoretical sensitivity by combining the concepts that have emerged from the data with the researcher's previous theoretical knowledge (Glaser and Strauss, 1967).

These basic rules have evolved considerably producing multiple types of grounded theory approaches. The first type is the Glaserian approach, which is the closest to the classical grounded theory. Glaser remained

consistent with the original grounded theory approach. He proposed two levels of coding; substantive and theoretical coding. According to Glaser (2002), the substantive coding aims to categorise data into as many categories as possible, while theoretical coding focuses on integrating the substantive codes to form a theory. The Glaserian approach is mainly based on emergence. Novice researchers find this approach challenging to be translated and applied in practice because emergence is highly reliant on theoretical sensitivity, which requires solid background in relevant theoretical directions (Kelle, 2010).

The second type is known as the Straussian approach. It was developed by Strauss as he was aware of the difficulties that face novice researchers in generating theoretical concepts. In 1987, he published a book titled "Qualitative Analysis for Social Scientists" to train students in grounded theory procedures (Corbin and Strauss, 2008). He proposed more systematic coding procedures and suggested three levels of coding: open, axial, and selective coding. Similar to the substantive coding in the Glaserian approach, open coding involves scrutinising the collected data to produce concepts. Axial coding involves aggregating concepts into categories. Strauss proposed a coding paradigm model to analyse the categories in terms of contexts, causation, intervening conditions, and consequences. Finally, selective coding sets the relations between the categories to develop a theory. Strauss elaborated his approach in a book titled "Basics of Qualitative Research", which was co-authored with Juliet Corbin (Corbin and Strauss, 1990).

A more recent type of grounded theory took a more flexible path. A famous example is the constructivist grounded theory developed by Cathy Charmaz (Charmaz, 2014). The advocates of this type assume that grounded theory constitutes of set of practices and principles; not a methodology. Charmaz argues that the grounded theory guidelines are flexible and can be applied with various methodological assumptions and approaches. She named her approach "Constructivist Grounded Theory" to acknowledge subjectivity

and highlight the role of the researcher's position, perspective, and interactions in the process of theory development (Charmaz, 2014).

In summary, "*there is no definitive grounded theory method*" (Corbin and Strauss, 2008:373). Despite the variations and contentions between the different types of grounded theory approaches, they all have the central elements of grounded theory. In the rest of this section I explain the main tenets of the grounded theory that have been adopted in this research.

3.2.2. The Grounded Theory Approach Applied in the Thesis

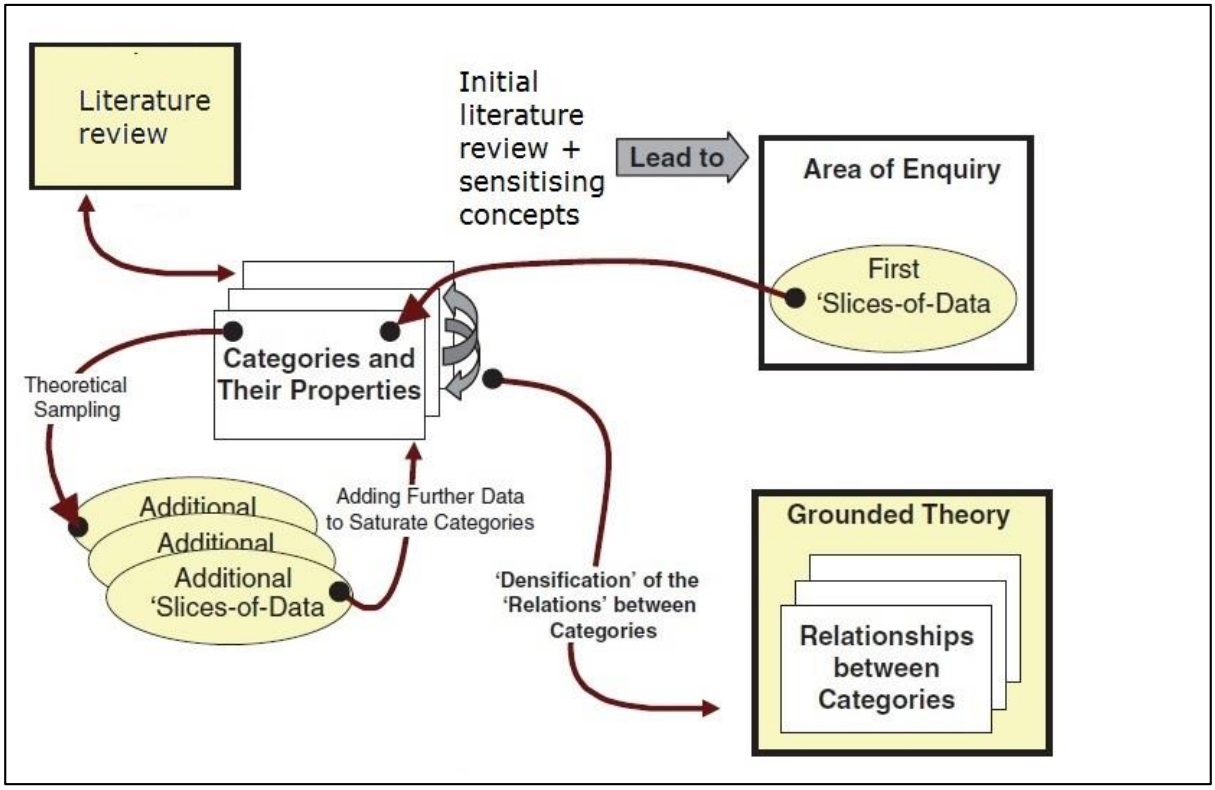
In this thesis, I adopt grounded theory approach throughout the research process including data collection, data analysis, and theoretical development. I have applied the basic tenets of grounded theory approach, as suggested by Urquhart et al. (2010), which are:

1. The research was not driven by pre-conceived assumptions or pre-formulated hypotheses. This does not mean that I have ignored the literature (Suddaby, 2006). Instead, an initial literature review was conducted and helped in clarifying the research topic and formulating provisional research questions. Besides, the initial literature review acted as sensitising concepts that guided the data analysis.
2. I have relied on multiple sources of data to provide an opportunity for triangulation.
3. 'Slices of data' of various types were selected by a process of theoretical sampling (Walsh et al., 2015). According to Glaser and Strauss (1967), theoretical sampling is the process of combining data collection and analysis to decide what data to collect next and where to find them. This allows the researcher to capture concepts as they emerge, which advances the development of theory. This is what distinguishes grounded theory from other qualitative research. In addition, in line with grounded theory tenets, data was not focused on individuals and organisations. It also included events, incidents, interactions, and consequences (Corbin and Strauss, 1990).

4. Constant comparisons were performed as the data were collected. Constant comparisons aimed to assign a common meaning to multiple data incidents (Locke, 2001). As concepts emerge and labelled they were compared to other incidents in data. This continued until the existing categories were saturated.
5. Since the main purpose of the grounded theory method is theory building, I had to maintain theoretical sensitivity, which refers to the theoretical awareness of the researcher. I have developed this awareness by frequently referring to the relevant literature (Goulding, 2002) and by staying close to data (Charmaz, 2014). In addition, as argued by Corbin and Strauss (2008), the more I was involved in data analysis, the more I developed theoretical sensitivity.

During data analysis, I have followed the Straussian coding phases (i.e. open, axial, and selective) and the Straussian coding paradigm. I have selected the Straussian approach because it provides a sign-posted procedure that guides the research (Seidel and Urquhart, 2013). It also provides analytical tools that uncover the context in which the target phenomenon occurs, and focus on related interactions and consequences (Corbin and Strauss, 1990), which fits the objectives of the research. The coding paradigm is flexible, and thus researchers can construct a coding paradigm that is consistent with their particular objectives (Kelle, 2010). Moreover, the Straussian approach allows researchers to more transparently report on the underlying research processes (Seidel and Urquhart, 2013). It is evident that the Straussian coding scheme became a common practice in the information systems discipline in general (Seidel and Urquhart, 2013). Besides, it has proven to be fruitful in OSS literature (e.g. Shah, 2006; Agerfalk and Fitzgerald, 2008; Feller et al., 2008; Shaikh and Henfridsson, 2017).

Figure 2: The Grounded Theory Approach



Adapted from Urquhart et al. (2010)

Figure 2 summarises the grounded theory approach followed in this research. The figure has been adapted from Urquhart et al. (2010). It illustrates that the research started with an initial literature review that lead to the area of inquiry and the collection of the first slices of data. Then, the research went through an iterative process between data collection, analysis, and literature review. This is considered as a central feature of grounded theory methodology. This process is further explained in sections 3.4 and 3.5.

3.3. Research Design: A Single In-Depth Case Study

In this thesis, I aim to understand how governance practices emerge in vertical OSS communities that are targeting the higher education (HE) sector. I have chosen Kuali as the target case. Kuali is a vertical OSS community that builds and maintains an enterprise (ERP) system for the HE sector. The uniqueness of Kuali community is best understood using a

case study research that allows exploring Kuali in its natural settings (Yin, 2014).

I have conducted a single in-depth case study using a grounded theory approach to understand "what is going on here", which is a central question in grounded theory (Corbin and Strauss, 2008). According to Eisenhardt (1989), a grounded theory that is developed from a case study is likely to have higher level theory. That is because, the iterative process of grounded theory moves the research beyond description and strengthens the research design as well as the validity of the findings. Besides, scholars in the field of organisation studies and information systems (e.g. Orlikowski, 1993; Urquhart, 1997; Boudreau and Robey, 2005) demonstrate that the case study method is compatible with grounded theory because case studies focus on contextual and processual elements that aid in conceptualising the phenomenon under study.

In the remaining of this section, I explain the HE settings with regards to adoption and implementation of OSS project. Then, I describe the organisational context of Kuali community.

3.3.1. OSS in the Higher Education Settings

The HE sector often falls into the build-buy dilemma when purchasing, developing, and maintaining information systems (IS) in general. The 'build' option allows universities to develop IS that are tailored to their requirements, which allows universities to gain a full control over the source code. This option is suitable for universities that afford leveraging developers, maintenance, and project management (Wheeler, 2003). With regards to the 'buy' option, it transfers the load of code development, maintenance, and management to a service provider. However, the 'buy' option is recognised for its high cost and low control over the source code from the university side. Besides, the software market does not meet the needs of the HE sector, mainly due to the idiosyncratic nature of the sector that includes unique business practices (Courant and Griffiths, 2006). The

business process workflow within a university is more complex than any other organisation and often not handled by a commercial software (Brooks, 2007).

OSS solutions introduced the concept of 'collaborate' to the software development process (Wheeler, 2003), which blends the benefits of 'build' and 'buy' options. OSS development process unbundles the software development and support granting universities more control over the future direction of the software (Wheeler, 2004). Besides, it promotes collaboration and pools resources from multiple universities, as well as firms, to build OSS applications that can be freely available to the wider community (Brooks, 2007). However, this cannot be achieved without disciplined and purposeful governance practices that align the OSS concepts with the objectives of the HE institutions (Wheeler, 2003).

The concepts of OSS are not new among HE institutes (Wheeler, 2003). In the 1960's and 1970's, the software development was carried out by scientists and engineers in academic institutions and research labs, where part of the development process was to freely share and exchange source code for modifications and improvements (Lerner and Tirole, 2002; von Krogh et al., 2012). The commercialisation of software is tracked back to the 1980's adding constraints to IS adoptions (von Krogh et al., 2012). Yet, this did not stop university efforts to continue sharing source codes.

Linux and Apache, one of the earliest OSS technologies developed by and entered to universities, were mainly located in data centres, used by technicians, and not exposed to students, faculty, and administrative staff members (Masson, 2011). Universities were concerned about the cost, reliability, and security aspects of developing and supporting IS for non-technical users (Wheeler, 2003; Masson, 2011). With the advancements in technology and the high demand on involving students and faculty in defining business requirements; especially in university specific applications like learning management systems, universities successfully

produced outstanding systems as OSS solutions (Masson, 2011). Examples of OSS produced by universities for universities are Moodle and Sakai[2]. This success encouraged universities to introduce OSS concepts to develop enterprise level IS, such as Kuali, the focus of this thesis. Universities found that OSS development practices are more effective when the developers are the users of the software (Courant and Griffiths, 2006).

Besides the internal drivers, such as cost, control, and performance (Courant and Griffiths, 2006; Benlian and Hess, 2011), that encouraged universities to initiate and adopt OSS projects, external factors played an essential role as well. Taking the United States (US) as an example, since it is one of the leading countries in promoting OSS in the HE sector (Courant and Griffiths, 2006), two main factors supported the diffusion of OSS among the American universities. First, the financial support provided by not-for-profit foundations which are dedicated to promote the activities of HE institution (Courant and Griffiths, 2006; Wheeler, 2007), such as Andrew W. Mellon and William and Flora Hewlett Foundations (Kuali Foundation, 2014). Second, the public policy in USA highly supports the OSS development. For example, in 1999, the US government recommended the President's Information Technology Advisory Committee to set a research strategy that uses OSS development as the new model for promoting the high end computing software needs in the United States[3].

The focus of this thesis is to understand the governance practices in OSS communities that are dedicated for and sponsored by the HE sector. Current studies represent how the OSS concepts created new forms of organisations and new governance practices where the individual-level is the main unit of analysis. By exploring Kuali community, I aim to understand how governance practices emerge and evolve in a vertical OSS

[2] Moodle and Sakai are open source learning management systems used in schools and universities.
[3] Developing Open Source Software To Advance High End Computing, October 2000: https://www.nitrd.gov/pubs/pitac/pres-oss-11sep00.pdf

community that challenges the taken-for-granted assumptions that are dominating the relevant literature.

3.3.2. Research Site: Kuali Community

Kuali is an OSS community dedicated to develop applications for the HE purposes by the HE employees. Kuali community was selected for its uniqueness as it challenges the presumptions that are dominating the existing literature. According to Kuali Foundation (2014), Kuali community was initiated in 2004 by Indianan University (IU) in USA when IU decided to replace its technically obsolete financial system. At that time, the world has just survived the millennium bug and various reputable ERP systems in USA have collapsed. The HE sector was under high pressure to reduce its expenditures. Therefore, IU refused to go through the painful experience of either building their own software or buying a commercial package. Instead, IU took a strategic decision by exploring a third option, which is collaboration. As a result, IU looked out for strategic alliances interested in developing a financial application tailored for the HE context.

IU joined the National Association of College and University Business Officers (NACUBO) to explore an innovative approach to build a financial system for HE institutes. IU and a group of universities started-off with implementing Kuali Financial System (KFS), which was based on the functionalities of IU's existing legacy financial application[4]. Kuali community was not a conventional OSS community. Instead, it followed a community-source approach that has opened-up the OSS code, but imposed restrictions on the OSS development process by implementing Kuali-specific framework called Rice. In other words, Kuali founders formed a gated vertical OSS community to control the code and align OSS development process with the HE settings in USA.

[4] IU's legacy finance system was technically obsolete. However, functionality-wise, it covered the basic requirements of the American universities.

In 2005, the success of KFS was appealing to the community of universities in USA, and thus provided a stepping stone to expand the scope of the community from developing KFS to the development of an open source ERP system. Thus, ERP modules were added, including the research and student systems. Each module was considered as an OSS project. Besides, non-American universities were also invited to join Kuali community in order to enrich Kuali projects. Subsequently, the founders realised that each project, university, and country had different requirements. Therefore, it was necessary to form a foundation to ensure consistent governance practices across different contexts.

Accordingly, in 2006, Kuali Foundation was established and received the not-for-profit organisation status to facilitate collaboration between various HE institutes, where developing OSS is one of their core objectives. Kuali founding partners comprised of Indiana University (IU), the University of Arizona, the University of Hawaii, Michigan State University, San Joaquin Delta Community College, Cornell University, and NACUBO. Kuali Foundation consists of board of directors and members. The membership is open to universities, research and development centres, and for-profit or not-for-profit organisations. The Foundation also sought to collaborate with Kuali commercial affiliates (KCA's) to provide complementary services to the community, such as planning, implementation, hosting, and supporting services.

Throughout the first decade, the implementation of Kuali products was carried out by HE employees and restricted by HE rules. These constraints hindered Kuali community from coping with the increasing demands and expectations of the stakeholders, especially with the advancements in the technology. Therefore, in 2014, Kuali community spun-off a for-profit company called KualiCo to improve and sustain Kuali products.

Kuali community maintains various OSS projects; however this thesis focuses on three major OSS projects; the Kuali Financial System (KFS), the

Kuali Student (KS), and the research administration system, which is called Kuali Coues (KC). These three projects have been chosen for their well-established communities and repositories of archival documents that illustrate Kuali governance practices.

3.4. Data Collection

OSS communities are known for their diversity and transparency of data for researchers (von Krogh and Spaeth, 2007), such as the publicly available email discussions, meeting minutes, and technical documentations. The existing literature on governing OSS communities is dominated by research studies focusing on the project level of the community, as illustrated by Crowston et al. (2012) in their recent literature review. The project level refers to the development team of a particular software. The current studies explain OSS governance as an emerging process that is determined by the communication patterns of the developers while building and maintaining the OSS (Demil and Lecocq, 2006; O'Mahony and Ferraro, 2007).

Due to the theory-building nature of this thesis, I have combined multiple types of data (Eisenhardt, 1989) to gain a coherent understanding of Kuali community and its governance practices. The data represents different levels of the community, including the executive level. The variation of data sources across levels and the uniqueness of Kuali community provided an opportunity to understand how governance practices emerged in Kuali. Sections 3.4.1 and 3.4.2 provide a detailed explanation of the secondary and primary data collection respectively. Section 3.4.3 describes the theoretical sampling procedure of data collection, and provides a summary of the collected data.

3.4.1. Secondary Data: Archived Documents and Videos

The data collection process began in October 2013 with an in-depth study of Kuali email archives, meeting minutes, developer's guides, and technical

regulations in order to gain familiarity with the community settings (Charmaz, 2014). These documents are available to the public; however a free Kuali online account was required for obtaining the access. Kuali archive goes back to 2010. It provides information on the organisational settings of the community, the day-to-day activities, governance structures, policies, and rules and regulations. During the data collection period (2013-2016), the archived data were distributed amongst 6 different technical artefacts, as shown in table 7. I have accessed these artefacts, and scanned the available data. Then, I extracted related materials, organised them in separate documents, and uploaded them into Nvivo software to facilitate the analysis process.

Technical artefact	**Description of artefact**	**Extracted Secondary data**
Kuali main website	Contains general information about Kuali Foundation, and Kuali products.	Background information about Kuali Foundation, and Kuali projects Meeting minutes
Kuali Jira	Issue and bug tracking system for all Kuali projects	Regulations with regards to reporting, following-up, and resolving issues
Kuali Wiki	A documentation system that includes Kuali project plans, meeting minutes, role descriptions, contribution regulations, and process workflows	Kuali governance structure Documentations: e.g. Kuali rules and regulations, KFS Developer guides
Kuali Information System (KIS)	A repository of all Kuali people, projects, teams, and organisations.	Information related to individuals and organisations involved in Kuali projects
Google Drive	Kuali shared documents	KFS email discussions
YouTube	Includes Kuali official videos, such as interviews with Kuali leaders, seminars, and sessions of Kuali Days[5].	Kuali Days' sessions during 2011, 2012, 2014, 2015

Table 7: The Secondary Data and their Corresponding Technical Artefacts

[5] Kuali Day: an annual conference organised by Kuali Foundation to gather community members in a face-to-face event.

Regarding the archived emails, they include conversations since 2010 only. Each conversation contains multiple threads. I have focused on the email discussions of the KFS team members because the KFS project is the frontrunner and has reached to a mature stage. Therefore, the KFS-related email discussions were rich in terms of diversity of topics. The email discussions were transparent and archived in Google Drive. They were categorised into technical and functional email folders. The technical folder consists of email conversations among KFS technical team members from various universities and commercial affiliates regarding KFS technical issues. The functional folder consists of email conversations among KFS functional team members discussing issues related the use of KFS functionalities. Besides, the functional folder also includes discussions related to new functional requirements.

The technical and functional folders have been manually scanned in order to extract a variation of topics and avoid replication. I have targeted email discussions that illustrate the code development process, community support, coordination, collaboration, and how the community handles the university-specific requirements. During the process of extracting email threads, it was noticeable that the technical issues are more diverse; there is less replication in the email topics. Therefore, I was able to find 45 technical email topics dated between the years 2013 and 2014 that cover different technical aspects of the KFS project. On the other hand, the functional topics are replicated. Functional users raise similar enquiries over time, and accordingly the search range was extended to the year 2012. As a result, 21 different functional email topics were extracted. They are dated between 2012 and 2015[6].

The email conversations also involve references to previously raised issues in Kuali Jira system. Therefore, these issues (e.g. enhancements, bug

[6] Appendix illustrates sample technical and functional email threads. Note that I have extracted the conversation and organised them in a way that shows the original email and the corresponding replies.

fixing) have been reviewed to understand how they were assigned, followed up, and resolved. In general, the email discussions reflect the communication patterns among Kuali members in the project level of the community. They also explain how individuals were coordinated through defined roles and responsibilities. It also illustrate the use of technical artefacts to facilitate coordination mechanisms. In addition, the email communications demonstrate the ethos of an OSS community, such as knowledge sharing and community support.

The secondary data also include the meeting summaries of Kuali Board members that were published on the main website of Kuali community. All available meeting summaries were selected; from January/2010 to December/2015. The meeting summaries illustrate the macro-level discussions of Kuali community, such as the financial aspects of Kuali, the HE policies in USA, planning, and the process of prioritising projects. Meeting summaries also include presentation slides introducing Kuali to newcomers, and summarising Kuali progress to the Board.

In addition, KFS developer rules and KFS regulations documents have been reviewed. These documents are published in Kuali website and Kuali Wiki. They contain instructions for the technical and non-technical members on how to join and contribute to the community. All secondary data were uploaded into Nvivo software to facilitate the analysis process.

Speakers (pseudonym)	Session	Event
S1	Empowering the Community	Kuali Days 2011, J.W. Marriott Indianapolis, USA
S2	Economics, Operations and Strategy of Kuali	Kuali Days 2012, Austin, USA
S3	Welcome	Kuali Days 2014, Day1, Session1, South Africa
S4	Introduction	Kuali Days 2014, Day1, Session1, South Africa
S5	Strategic overview of Kuali	Kuali Days 2014, Day1, Session1, South Africa

S6	Stellenbosch University's journey to a Kuali decision	Kuali Days 2014, Day1, Session4, South Africa
S7	Kuali Collaboration in South Africa	Kuali Days 2014, Day1, Session4, South Africa
S8	Kuali's mission & progress in 2015	Kuali Days 2015, Austin, USA
S9	An overview of Kuali product offerings	Kuali Days 2015, Austin, USA
S10, S11, S12, S13	Explaining the Economics, Operations, and Strategy of Kuali for Higher Education	Kuali Days 2015, Austin, USA

Table 8: List of Kuali Videos

The final type of secondary data was based on official Kuali videos that are published in YouTube. The videos are 20 to 45 minutes long and represent keynote speeches of Kuali administrative and technical leads during various sessions of Kuali days, as summarised in table 8. The speakers are core personnel in American and non-American universities and companies who played an essential role in building Kuali community, defining the core values of Kuali, and reconfiguring the rules and regulations as the community evolved. The content of the videos complemented the data gathered from the interviews, especially with regards to the historical and background information about building Kuali community, and how the governance practices evolved over time. The videos were transcribed and uploaded to Nvivo as well.

3.4.2. Primary Data: Interviews

The primary data was based on 16 semi-structured in-depth interviews that were conducted via the Internet using Skype and Google Hangouts applications. Each interview lasted for 30 to 60 minutes, and was recorded with permission. The interviewees are categorised into administrators and technicians who are, at the time of data collection, employees in Kuali Foundation, Kuali Commercial Affiliates, KualiCo, or the contributing universities. The administrator category includes members responsible for the functional side of Kuali projects, such as executive directors, strategic

advisors, and business analysts. The technicians represent those who are responsible for the technical side of the projects, such as programmers, and IT consultants. Table 9 presents descriptive information about the interviewees. Some of the interviewees have multiple roles in the community; however table 9 displays their job title as shown in Kuali Information System.

Interviewee (pseudonym)	**Organisation**	**Team**	**Job title**
Admin1	Kuali Foundation	Kuali Foundation	Executive Director
Admin2	Kuali Foundation	Kuali Foundation	Strategic Advisor
Admin3	University (USA)	KC	Director, Research and Information Systems
Admin4	University (USA)	KFS	Senior Consultant to the CIO
Admin5	University (USA)	KFS	Financial Management Services
Admin6	University (Canada)	KS	CIO UT (KS)
Admin7	University (USA)	KC	Senior Director, Research Partnership Services
BA1	KualiCo	KFS	Business Analyst
BA2	University (USA)	KFS	Business Analyst
PM1	KualiCo	KFS	Product Manager (previous KFS functional member in a university)
Tech1	KualiCo	KFS	IT Consultant
Tech2	KualiCo	KFS	Analyst
Tech3	University (USA)	KFS	Applications Programmer
Tech4	University (South Africa)	KS	Senior Consultant (KS)
Tech5	KualiCo	KC	Research Compliance Product Owner
Tech6	University (Canada)	KS	Project manager (KS)

Table 9: Details of Interviewees

The interviews were conducted in two rounds that took place between 03/07/2015 to 03/08/2015 and 15/12/2015 to 28/03/2016 respectively.

The interviewees were identified during the process of analysing Kuali archived documents. The regulation documents illustrate that the Kuali Foundation is the governing body of the community, and accordingly the employees of the Foundation played an essential role in the governance process. The Foundation included (at the time of data collection) 14 employees; 12 of them were Board directors and officers that were elected in a yearly basis, and they were also holding administrative positions in their corresponding universities. The remaining 2 employees, which have been chosen as interview participants, were purely representing the Foundation. They are aware of the governance practices since the inception of Kuali. The first participant is the executive director of Kuali Foundation, who then became the gatekeeper for the second round of interviews. The second participant is the strategic advisor of Kuali Foundation, who is also one of the community founders. The rest of the participants of the first round were identified from the email discussions.

During the first round of interviews, I ensured to focus on the context of the American universities as they form the majority of the community. The initial coding process revealed that the target Kuali projects (i.e. KFS, KC, KS) were designed with the American HE settings in mind. However, there were some unique requirements determined by the federal governments of the states. Therefore, representatives from universities of different states within USA have been selected to understand how Kuali community responded to those specific requirements.

In addition, the analysis of the first round of interviews revealed that the context of country has an influence in the governance practices, and accordingly the second round of interviews included representatives from South Africa (SA) and Canada. Universities from these countries were chosen in particular because they have implemented at least one Kuali project of interest (i.e. KFS, KS, KC) and went through the process of implementation and post-implementation, and thus experienced the emergence and reconfiguration of various governance practices. Besides,

they have representatives in Kuali Board, and accordingly have influence on the future direction of the projects. In addition, the corresponding universities in SA and Canada have implemented KS project, which has been developed from scratch and had a very slow progress in comparison to the KFS and KC due to the complex user requirements. Accordingly, they have faced different challenges with regards to governance (this will be further explained in chapter 5).

Moreover, the reason for selecting SA is because it has different HE requirements in comparison to the context of USA, which is dominating the design of Kuali projects. An interview was conducted with a senior technical member from SA that was present during all implementation stages of KFS and KS projects and is experienced in both the administrative and technical aspects of the projects. Furthermore, Canada was chosen because it has similar educational settings with USA, but different governmental policies that had influenced the governance of Kuali projects in Canada. Interviews have been conducted with a senior administrative and a senior technical member of KS project in Canada. To sum up, the objective of choosing SA and Canada in particular was to cover different user perspectives and different contexts in order to enrich the research data.

The interviews were semi-structured based on interview guides. I have prepared a different interview guide for each participant. Figure 3 represents a sample interview guide[7]. As recommended by Charmaz (2014), the interview guide is a detail of all possible questions that are related to the research questions to assist the researcher in being spontaneous during the actual interview. Therefore, I have prepared a guide for each interviewee. The guides mainly included general questions related to the interviewee's day-to-day activities, the workflow process, the communication patterns with geographically dispersed team members, and the design and use of the technology and how it has influenced the

[7] Appendix A includes more samples of interview guides

coordination and control of the projects. Besides, each guide included question that were specific to the role of the interviewee in Kuali community. The actual interview included more questions that were inspired by the interviewee's answers. The interview provided real-time and retrospective data about Kuali. They provided different perspectives on the previously collected secondary data (Gioia et al., 2013). The interviews were transcribed and uploaded to Nvivo.

Figure 3: A Sample Interview Guide

Interview Guide-BA1

1. Could you please describe your current roles and responsibilities in Kuali community?
2. Did you have different responsibilities when you first joined in?
3. Kuali projects are available for free on Kuali website, from a functional lead perspective, why to bother join a community and pay annual membership fees to get the code?
4. With regards to new functional requirements, or enhancements, what is the process (work flow) within your university to accept or reject the requirements?
5. Are there occasions were you have rejected new functional requirements?
6. What is the work flow for adding specifications into KFS project plan? How do they get approved by Kuali Board?
7. Are these workflows the same since you have joined the community or they have been changed?
8. When you have multiple requirements/issues, how do you prioritise them? What comes first?
9. Does your university have unique functional specifications that do not attract the interest of the community?
10. So, do you have KFS features that are specific for your university?
11. You university is one of the founders of Kuali community. How does this impact your position within the community? I mean do you have a stronger influence on Kuali future direction? Or you are considered as a reference due to your experience?
12. Did you face situations where you had to change some business rules to cope with KFS release or technical requirements?
13. Monitoring is an essential concept in governance. How do you monitor the behaviour of individuals within the university to check whether they are adhering to Kuali values and practices?
14. Testing is essential for quality assurance. Could you please summarise the tasks you perform with regards to testing?
15. Was KFS implementation project entirely performed by the university employees, or with the assistance of KCA's? If KCA's, what was their role?
16. How far do you rely on the community in providing support?
17. As a member of the road map committee, how did the priorities of Kuali have changed throughout the last decade?
18. You have also been the director of the communication and outreach. What were your main responsibilities?
19. Is there anything else you think is useful for my research on governance and I did not cover in the interview?

3.4.3. Theoretical Sampling

As the research was following a grounded theory approach, the data collection and analysis were performed simultaneously in an iterative

manner. The data collection was guided by theoretical sampling process. The research started with an initial set of data that acted as a point of departure (Charmaz, 2014). The initial sample was based on Kuali secondary documents, which were used to understand the structure of the community, the different roles and responsibilities, and the day-to-day activities. In particular, I have started by analysing the regulation documents and email discussions because they illustrate the governance rules in principle and in practice respectively. The regulation documents explicitly state the roles and responsibilities of individuals and organisations in the community. On the other hand, the email discussions illustrate how the rules and regulations were applied in practice. This was evident in how the members respond to the different enquiries raised in the emails.

However, the initial sample did not explain how governance practices changed over time. Therefore, I then started looking into the meeting summaries of Kuali Board. The Board consists of representatives from the partner universities. The meeting summaries provide valuable information on the project management of all Kuali projects, such as planning, managing resources, and distributing tasks. They also demonstrate the process of prioritising projects, and balancing the specific university-related needs with the common requirements of the community. This has provided an overview on the overall context of the Foundation on one hand, and the changing requirements of Kuali projects on the other hand. Appendix B includes samples of meeting summaries.

Analysing the meeting summaries revealed that they lack sufficient historical information. This means, the meetings do not explain the circumstances that surrounded the inception of Kuali and how the governance practices emerged in the first place. This was important because Kuali represents a unique form of vertical OSS communities that is targeting the HE sector. Therefore, it was necessary to interview administrative and technical members to fill in these gaps.

As described in section 3.4.2, the interviews were conducted through two rounds. Analysing the first set of interviews revealed gaps in the data, which prompted for more interviews. By the end of the second round of interviews, I have found that there are official videos of core personnel in Kuali that will complement the previously data gathered from the interviews.

Research data	**Collection period**	**Description**
KFS functional emails	Oct/2013 – Mar/2015	-21 email discussions, total word count: **9496** -Dated from Aug/2012 to Mar/2015
KFS technical emails	Oct/2013 – Mar/2015	-45 emails discussions, total word count: **17650** -Dated from Jan/2013 to Dec/2014
Documentations (KFS rules and regulations)	Jan/2015	-24 documents, total word count: **23033**
Kuali Board meeting summaries	Feb/2015 – Dec/2015	-51 meetings, total word count: **46733** -Dated from Jan/2010 to Dec/2015
Interviews	Jul/2015-Mar/2016	-16 semi-structured, 30-60 mins long
Videos	Mar/2016	-7 videos, transcriptions' word count: **42828**

Table 10: A Summary of the Research Data

Further data were collected based on the analysis process in order to enrich the research data and support the emergent theoretical concepts. The process of theoretical sampling continued till the research reached to a point of saturation, which is the point where data did not provide additional theoretical insights (Locke, 2001; Charmaz, 2014). Table 10 illustrates a summary of the collected data. In the next section, I provide a detailed explanation of the analysis process.

3.5. Data Analysis

The secondary and primary data were analysed through coding, constant comparison, and theoretical sampling. In grounded theory research, coding

is the process of breaking down the data into manageable segments, conceptualising them, and putting them back together in a different way to develop a theory (Corbin and Strauss, 1990). As explained in Section 3.2.2, I have adopted the Straussian approach to coding, which involved open, axial, and selective coding.

Figure 4: Summary of the Data Analysis Process

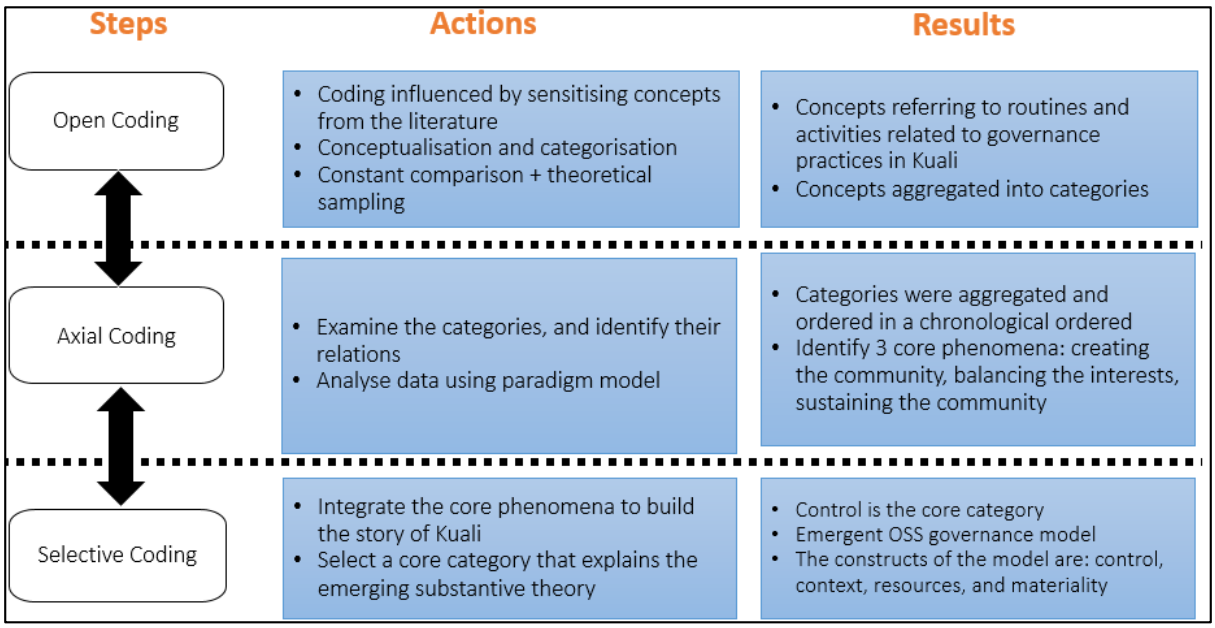


Figure 4 summarises the data analysis. Although the analysis was an iterative process, the figure illustrates a linear transition for clarification purpose. The following is an explanation of the three coding phases.

3.5.1. Open Coding

Open coding was the first analytical step. It was an interpretive process to assign meaning to raw data (Corbin and Strauss, 1990; Corbin and Strauss, 2008). Open coding was sensitised by the initial literature review. In particular, it was mainly driven by four sensitising concepts. The first sensitising concept was the definition of OSS governance as declared by Markus (2007), which states that OSS governance is "*the means of achieving the direction, control, and coordination of wholly or partially autonomous individuals and organizations on behalf of an open source software development project to which they jointly contribute*" (p.152).

This highly cited definition has been selected because it is comprehensive and summarises how OSS governance is explained in the relevant literature in terms of structure, practices, and culture (von Krogh et al., 2012; Di Tullio and Staples, 2013).

The second sensitising concept is the consensus in the literature with regards to the fluidity and dynamicity of OSS communities. However, at the same time the current accounts explain governance by focusing on fixed structures that facilitate the collaboration among dispersed individuals, neglecting the issues of fluidity and dynamicity (Aaltonen and Lanzara, 2015). The third sensitising concept is the emphasis on the role of technology as a mediator or as an end product. This view does not provide sufficient explanation on the role of the material characteristics of the technology in coordinating the dispersed community resources (Faraj et al. 2011). The fourth sensitising concept was based on the call for redirecting the scholars' attention towards governing vertical OSS communities (e.g. Crowston et al., 2012). These communities, according to the literature review, have different settings in comparison to those described in the literature, and thus require further exploration. These sensitising concepts were the "*points of departure for studying the empirical world while retaining the openness for exploring it*" (Charmaz, 2014:30-31).

The process of open coding involved two steps: conceptualisation and categorisation. The main purpose of conceptualisation was to develop concepts (Corbin and Strauss, 1990), where concepts refer to the underlying meaning of the data and the patterns that emerge from them (Goulding, 2002). The aim was to develop concepts that explain how Kuali community governed the collective effort, and how the governance practices evolved overtime. According to Corbin and Strauss (1990), concepts are the main unit of analysis during the coding process. They can take the form of an event, incident, object, and/or interaction. I have developed concepts in the form of salient incidents and interactions that were related to governing the collective effort in Kuali community.

Conceptualisation first required breaking down the qualitative data into manageable pieces to examine them, and compare them for similarities and differences (Corbin and Strauss, 2008). Nvivo software provided a convenient way to perform a line-by-line reading of the secondary and primary data. It also facilitated the process of segmenting the data and labelling them with meaningful codes. The transcriptions of interviews and videos were then conceptualised using the terms used by the participants. The archived documents were treated as texts and were situated in their context (Charmaz, 2014). The analysis of the archived documents involved identifying key individuals, organisations, and processes related to each document. Individuals were classified by their role in the community, their work place, and previous experiences. Organisations were classified by type (university or firm), location (i.e. country), and their role in Kuali community. Therefore, the concepts that emerged from analysing archive documents used terms that are close to the context of the document.

During conceptualisation, besides analysing the primary and secondary data, I have also focused on the design and use of the technology in Kuali community. From the interviews, data revealed that Kuali did not inherit the conventional OSS development process where volunteers are invited to co-develop the OSS code. Instead, the OSS license, the structure of the code, and the coding regulations were continuously changed to meet the change in the context of Kuali. Moreover, I have observed how the technical artefacts were designed and utilised by different user groups in different Kuali projects. Although the technical artefacts were standardised among all Kuali projects, their features and use were altered based on the context. This is further explained in the findings chapters (chapters 4, 5, and 6).

Initially, conceptualisation generated provisional concepts that were gradually refined as further data were collected and coded (Goulding, 2002). Constant comparison was the primary analytical tool during this step. Incoming data were constantly compared with the emergent concepts in order to identify gaps in the data, which informed further data collection.

Besides, the emergent concepts were also repeatedly compared across incidents, participants, time, and context (Locke, 2001) to understand governance practices in Kuali. This was facilitated through the use of Nvivo, which provides set of tools that assist in searching for particular text or patterns. After an iterative process of creating provisional and comparative list of concepts, conceptualisation produced concepts that were descriptive and close to data.

The second step of open coding was the categorisation. The categories are higher in level and more abstract than concepts (Corbin and Strauss, 1990). Categories pull together the concepts into theoretical framework (Goulding, 2002). This involved sorting, synthesising, and aggregating the concepts into categories that correspond to OSS governance. Constant comparison was applied again at this stage by going through a recursive process of comparing concepts to data in order to understand the emergent patterns with regards to governing vertical OSS communities. Table 11 illustrates the outcome of open coding. I have chosen 22 salient concepts that made the most analytical sense (Charmaz, 2014). These concepts were aggregated into 8 categories corresponding to the governance of Kuali community.

Concepts	**Categories**
Considering the context of HE in USA	Contextualisation
Considering the context of Kuali projects	
Considering the context of international universities	
Choosing suitable OSS license	Controlling the code
Fulfilling HE needs	
Gating the community	
Selecting resources	

Membership and partnership	Structuring the community
Assigning roles	
Who can contribute	Setting properties of OSS code
What is contributed	
How to control contributions	
Issues with contribution model	
Balancing university and community needs	Managing divergent interests
Adjusting local workflow to Kuali workflow	
Coordination and communication	Facilitating collaborations
Organising meetings	
Using standard technology	
Creating KualiCo	Restructuring the community
Adjusting community roles	
Adjusting technical artefacts	Reconfiguring coordination and communication
Adjusting means of communication	

Table 11: The Outcome of Open Coding

As shown in table 11, the open coding produced 8 categories. The first category, *contextualisation,* is not directly related to OSS governance. Instead, it is about be sensitive to the evolving context of Kuali community, which outlines the overall direction of the community. The remaining 7 categories refer to the salient governance practices of Kuali community that were evident in Kuali during the period between 2004 and 2016. These categories highlight the dynamicity of the community and the materiality of technology.

Open coding revealed that the inception of Kuali community was highly influenced by the previous experiences of the American universities with respect to ERP projects. In addition, the conventional OSS governance practices that were explained in the relevant literature were reconfigured to align with the HE settings in USA. Therefore the meaning of openness did not refer to the complete transparency and accessibility, rather the community decided to draw boundaries around the OSS code and the OSS development process to meet the needs of the HE sector in USA.

In summary, the process of open coding segmented the data into concepts and categories. The next coding phase, i.e. axial coding, reassembled the data back by locating and linking actions and interactions within a framework that gave it meaning and explained why interactions were occurring and what consequences were anticipated (Corbin and Strauss, 2008).

3.5.2. Axial Coding

The aim of axial coding was to reassemble the data that was fractured during open coding in order to develop a substantive theory. It was mainly about looking for clues on how the categories fit together (Corbin and Strauss, 2008). According to Goulding (2002), open code categories are not standalone. Instead, their properties are connected and can be aggregated in a hierarchal, linear, or recursive form. Therefore, the axial coding phase started by examining the 8 open code categories in detail, and how they are related to each other. This was done to identify the core phenomena in Kuali community. The following is an explanation of how and why the categories (*in italics*) were linked in a linear form.

Based on the analysis of Kuali data, Kuali community was formed in 2004 to develop Kuali Finance System (KFS) for the HE institutes in USA. The inception of Kuali was preceded by *contextualisation*. This practice set the community goals based on the requirements of the context. Accordingly, Kuali community aimed to build a university-specific OSS community to

prevent non-universities from influencing the functionalities of the finance system. Therefore, *controlling the code* emerged as the salient governance practice.

As the open source KFS evolved, it attracted the interests of other departments within the universities (i.e. other than the finance department). Therefore, during 2005, the scope of the community was expanded from developing a finance system to the development of a full ERP-suite. The expansion required inviting new members to the community in order to enrich the community with diverse resources. Therefore, Kuali Foundation was formed in 2006 to facilitate governance practices. The governance practices that have emerged were *structuring the community, setting the properties of the OSS code, managing divergent interests,* and *facilitating collaborations.*

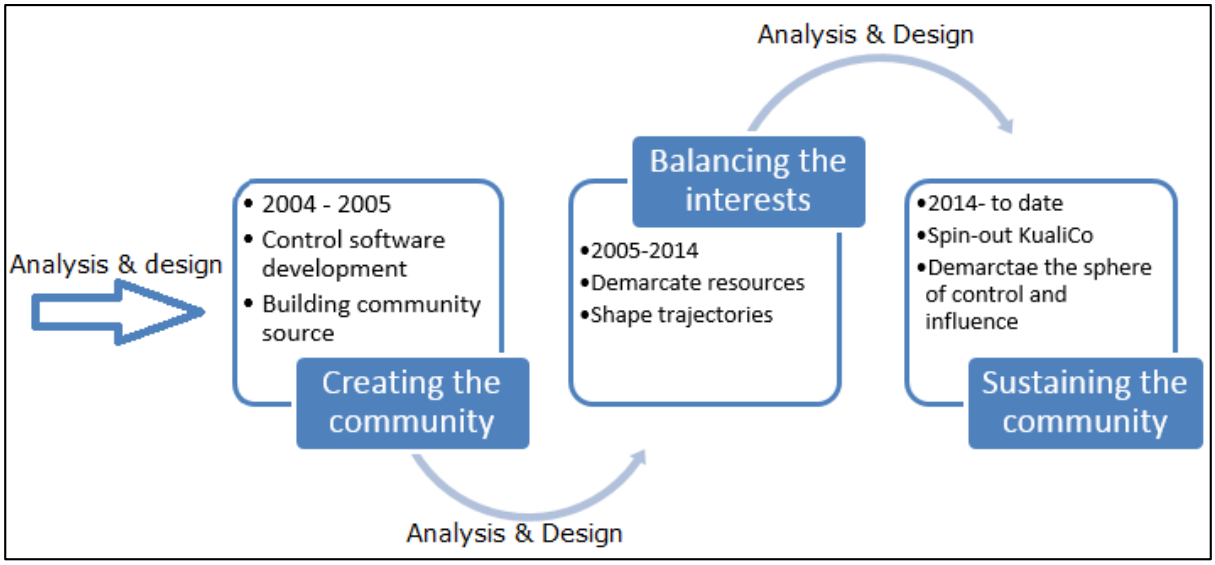
However, the community faced various administrative and technical challenges to handle the needs of the diverse stakeholders. Therefore, in 2014, KualiCo, a for-profit company, was formed to take over the development of Kuali projects. KualiCo was also granted the authority to control the direction of Kuali projects. The decision to introduce market considerations within the community changed the boundaries of Kuali community. KualiCo was introduced as a key player, which included and excluded actors, actions, and rules. Accordingly, the community was governed by *restructuring the community,* and *reconfiguring coordination and communication.*

From the above analysis, it means that the boundaries of Kuali were not fixed. Instead, the community boundaries evolved as the scope and context changed over time. Thus, OSS governance practices changed as the community developed. It was noticeable from the data that Kuali went through three governance phases. Each phase emerged from a boundary decision. In other words, each governance phase emerged from decisions

to change the boundaries of the community in order to accommodate the changes in the surrounding context.

The boundary decisions are: creating the community, balancing interests, and sustaining the community. The governance phases were explained in details by the interviewees, and also explicitly mentioned in the secondary data (e.g. meeting minutes, emails). Each phase was preceded by the processes of analysis and design (i.e. contextualisation) to determine the new scope and objectives. Each phase experienced a set of activities and routines that correspond to governance practices. This is summarised in figure 5.

Figure 5: Kuali Life Span



Source: Author's Own Figure

Accordingly, the concepts and categories were aggregated and ordered chronologically to represent the three governance phases of Kuali, where the governance phases are the core phenomenon. As shown in table 12, the core categories correspond to the boundary decisions that moved Kuali to different governance phases. The categories correspond to the salient governance practices of each governance phase. The concepts refer to the routines, actions, and incidents that are related to each governance practice. Notice that the category *contextualisation* is not included in the table. That is because it refers to the ongoing processes of analysis and

design that were evident in Kuali community to realise the dynamicity of the context. This is further explained in the following lines.

Concepts	Categories	Core phenomena (Boundary decisions)
Choosing suitable OSS license	Controlling the code	Creating the community
Fulfilling HE needs		
Gating the community		
Selecting resources		
Membership and partnership	Structuring the community	Balancing the interests
Assigning roles		
Who can contribute	Setting properties of OSS code	
What is contributed		
How to control contributions		
Issues with contribution model		
Balancing university and community needs	Managing divergent interests	
Adjusting local workflow to Kuali workflow		
Coordination and communication	Facilitating collaborations	
Organising meetings		
Using standard technology		
Creating KualiCo	Restructuring the community	Sustaining the community
Adjusting community roles		
Adjusting technical artefacts	Reconfiguring coordination and communication	
Adjusting means of communication		

Table 12: Identifying Core Phenomena

After identifying the core phenomena, I have organised the results of axial coding using a coding paradigm model that is influenced by the Straussian paradigm. A coding paradigm is an analytical tool to understand the circumstances that surround the core phenomenon, and therefore enrich

the analysis (Corbin and Strauss, 2008). The Straussian paradigm, as explained by Corbin and Strauss (1990), comprises of the main properties of a core phenomenon, which are:

- Phenomenon: refers to the central idea of event.
- Context: refers to the properties of the phenomenon.
- Strategies: are the actions and interactions that manage or respond to the phenomenon.
- Conditions: either refer to the incidents that cause the phenomenon to occur or intervene the actions/interactions.
- Consequences: are the results of strategies when dealing with the phenomenon.

The core phenomenon represents the category while the properties of the phenomenon become sub-categories. The sub-categories address the where, how, why and with what consequences the phenomenon occurs to further conceptualise the phenomenon. It is important to point out here that the Straussian coding paradigm is only a tool and not a set of directives. Therefore, adjusting the coding paradigm to fit the data rather than trying to force the data into predetermined categories is a common practice by researchers (Urquhart 1997). Accordingly, as shown in table 13, I have adjusted the properties in a way that is consistent with the data and the objectives of the research. Then, I have applied the paradigm to each core phenomenon.

Core phenomenon	**Building community**	**Balancing interests**	**Sustaining community**
Conditions	Analysis and design: - Considering the context of HE in USA	Analysis and design: -Considering the context of Kuali projects -Considering the context of international universities	Analysis and design: -Considering the context of international universities

Governance practices	-Controlling the code	-Structuring the community -Setting properties of OSS code -Managing divergent interests -Setting means of collaboration -Setting properties of technical artefacts	-Restructuring the community -Reconfiguring coordination and communication
Consequences	-Building a community source -Universities are the controllers -Functionally driven community -Prompted the need to build a full open source ERP suite	-Tight control -Projects struggle with restrictive Rice -Projects are progressing slowly -Triggered the need to rethink Kuali strategy and spinout KualiCo	-KualiCo is the controller -Universities are influencers, not controllers -The community source model was transformed to a vendor-customer model

Table 13: The Paradigm Model of the Core Phenomena

The core phenomena corresponds to the 3 major boundary decisions in Kuali life. Each boundary decision delineated the boundaries of Kuali community by determining who/what is in and out of the community, the actions, the scope of the community, and the required resources. Accordingly, each boundary decision started a new governance phase because it included different objectives, community settings, and bundle of governance practices.

The properties of the boundary decisions are conditions, governance practices, and consequences. The conditions refer to the processes of analysis and design. These are ongoing processes to realise the changes that occur to the surrounding context, and accordingly trigger the conditions that cause the boundary decision. The governance practices are the actions and interactions that respond to the boundary decisions. The consequences refer to the main outcomes of the governance practices. Mainly consequences contribute to the emergence of a new conditions, and thus trigger new boundary decisions.

The coding paradigm provided insights to answer the research questions that evolve around dynamicity and materiality. Arranging the core phenomena in a chronological order and explaining them in terms of conditions and consequences represent the dynamic nature of Kuali community and how the governance practices emerge and evolve. With regards to materiality, it was evident the setting the properties of the OSS code and the technical artefacts emerged as salient governance practices in Kuali community to control the coding process and to facilitate collaboration. This means that materiality was inseparable from the governance practices, as will be further explained in chapters 4, 5, and 6.

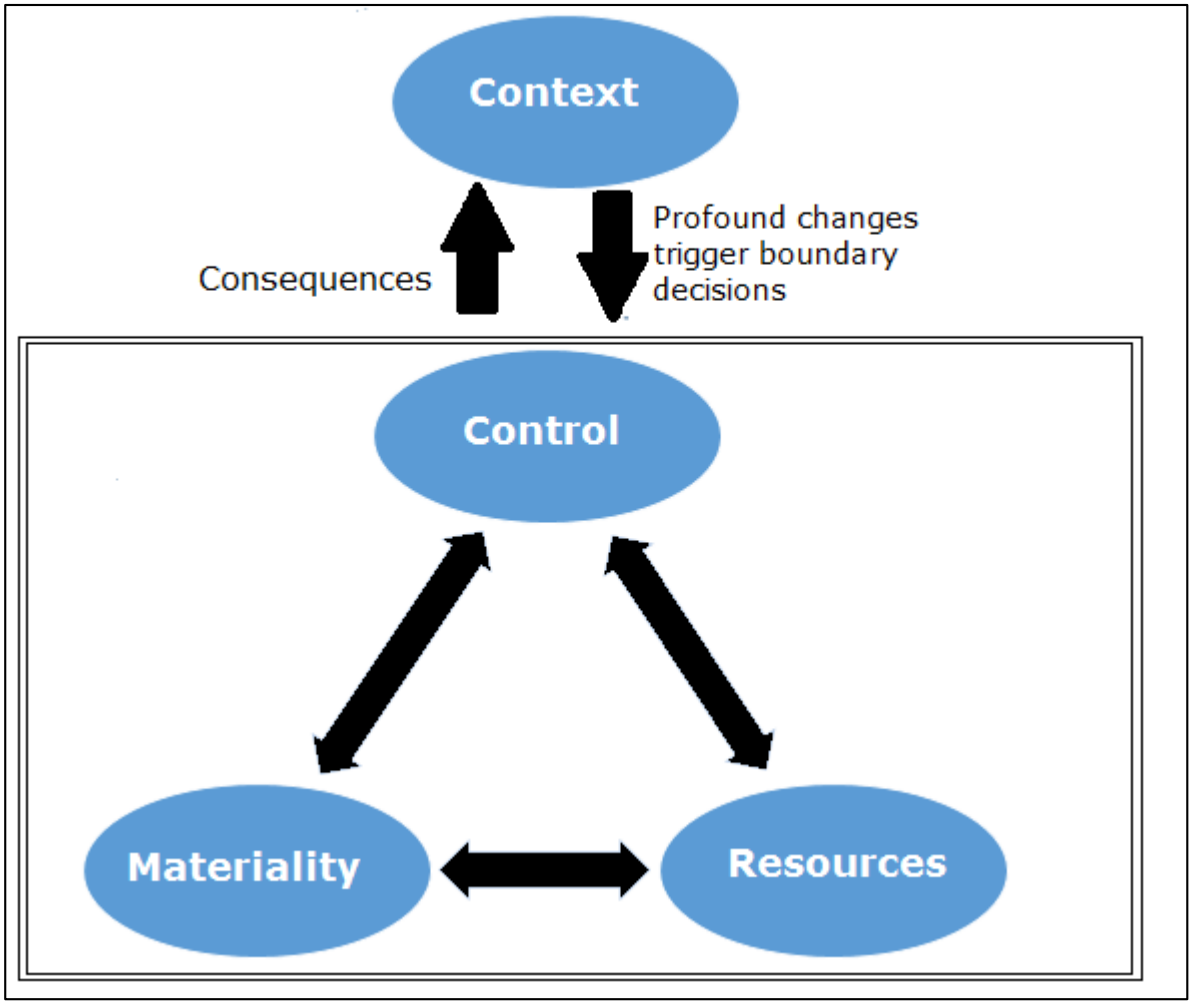
The coding paradigm also assisted in the process of articulating the developed substantive theory (Locke, 2001). The completion of the paradigm model provided a theoretical structure, which was the building block for the selective coding phase. However, before moving to the selective coding phase, it was necessary to ensure that the research has reached to a point of theoretical saturation (Goulding, 2002), which means that further data collection does not provide insights to construct new categories or refine existing ones (Charmaz, 2014). Open and axial coding were iterative processes. They were brought to an end when the emerging categories were sufficient to construct a coherent story about the phenomenon under study (Locke, 2001; Creswell, 2007). Chapters 4, 5, and 6 narrate the story of Kuali and provide data illustrations on Kuali governance phases and their properties.

3.5.3. Selective Coding

The selective coding was the final phase of data analysis. It aimed to explain the relationship between the core phenomena that emerged from the axial coding phase. The first step of selective coding was to identify the core category that integrates the core phenomena into a holistic theory. According to Corbin and Strauss (1990), the core category has to be central and appears frequently in the data. This means that all other major

categories can be related to it in a logical and consistent way. In addition, the name or phrase used to describe the core category should be sufficiently abstract. The core category should have the analytical power to pull all the other categories together to form contextual explanation of OSS governance rather than descriptions. Identifying the core category required analysing the paradigm.

Figure 6: The Development of Substantive Theory



Source: Author's Own Figure

Referring to the coding paradigm represented in table 12, the core phenomena corresponds to the three boundary decisions of Kuali. These decisions emerged as a response to the changes in the context, as shown in the first row (i.e. condition) of the table. The salient governance practices of Kuali challenge the taken-for-granted assumptions in the literature of OSS governance. Although the OSS code of Kuali projects was publically

available, the OSS development was gated and limited to the HE sector. It is evident that the actions and interactions in each phenomena evolves around controlling the code to maintain university-driven OSS products. This means that creating Kuali as a vertical OSS community was a boundary decision to control the code from being influenced by non-universities. Table 12 also suggests that the second and third boundary decisions were necessary to control the community on one hand, and enable the growth of the community on the other hand.

Based on that, *control* has emerged as the core category for the substantive theory. During axial coding, it was evident that controlling the code in Kuali was the main desire to form the community in the first place. It continued to be the main driver of governance as the community evolved. *Control, context, resources,* and *materiality* are the main constructs of the boundary decisions. They interact with each other to change the boundaries of the community, and thus govern the collective effort. Figure 6 illustrates the emergent OSS governance model that illustrates the substantive theory. It is explained in details in chapter 7.

3.6. Summary

In this chapter, I have clarified the research methodology. I started this chapter by explaining the rationale for selecting a grounded theory approach as the research framework, and a single case study as the research design. Then, I provided details on the different types of data sources and the data collection method. Finally, I explained in details the data analysis process. I have illustrated the process of moving from data to a substantive theory with regards to OSS governance.

Chapter	Content	Related to
4	Core phenomenon: Creating the community	Open and Axial coding
5	Core phenomenon: Balancing interests	Open and Axial coding
6	Core phenomenon: Sustaining the community	Open and Axial coding
7	Theory development	Selective coding

Table 14: Summary of the Next Chapters

Table 14 summarises the content of the next chapters. In chapters 4, 5, and 6 I explain through empirical evidence the properties of the core phenomena. Each chapter corresponds to a phenomena, which means each chapter explains a boundary decision that moved Kuali community to a new governance phase. For each boundary decision, I illustrate the conditions, interactions (i.e. governance practices), and consequences. Then, in chapter 7, I elaborate on the selective coding phase by explaining the emergent OSS governance model. I also discuss how the emergent model is related to the existing literature. In chapter 7, I also explain the evaluation criteria of the research.

Chapter 4: Kuali Governance Phase 1: Creating the Community

4.1. Introduction

In this chapter, I illustrate through empirical evidence that creating the community was a boundary decision to create a vertical OSS community, and thus the first governance phase was initiated. This chapter is structured as per the paradigm model explained in section 3.5.2. This means, I will begin this chapter, in section 4.2, by clarifying the conditions that triggered the first boundary decision. Then, in section 4.3, I illustrate through evidence how the governance practices emerged in Kuali. In section 4.4, I illustrate the consequences of creating the community. Finally, the chapter is concluded in section 4.5.

Although Kuali community has formally started-off its activities in 2004, the publicly available archived data were logged starting from 2010 only. This means that the archived data do not provide sufficient information on the inception of Kuali community. Therefore, the sources of data in this chapter are mainly interviews and Kuali official videos to provide historical and background information about Kuali community. As mentioned in chapter 3, the interviewees are administrative and technical members of Kuali community. The official videos are for various sessions of Kuali Days in 2011, 2012, 2014, and 2015, where the speakers cover historical information about Kuali.

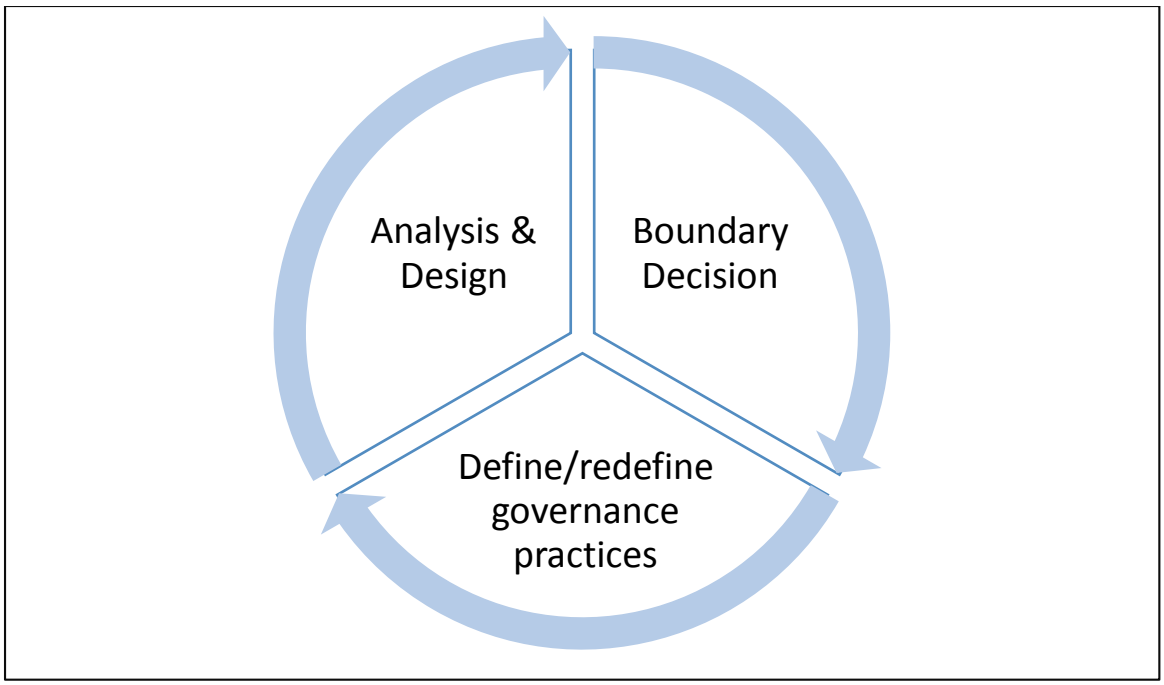
4.2. Conditions of Phase 1: Considering the Context of HE in USA

The conditions that triggered the need to create Kuali community emerged from the processes of analysis and design. One of the major findings of this thesis is that each boundary decision in Kuali community was preceded by the processes of analysis and design, i.e. contextualisation. These processes were not explicitly considered as formal practices in conventional

OSS communities (Crowston et al., 2012). They have emerged due to the transition from horizontal to vertical domains (Rolandsson et al., 2011). As discussed in chapter 2, in horizontal domains, the software requirements and the design issues are generic and known to the developers. On the other hand, in a vertical domain-specific OSS communities, the software requirements rely on the context and the stakeholders (Fitzgerald, 2006).

As illustrated in figure 7, the processes of analysis and design observe and assess the overall context of Kuali community. These processes trigger boundary decisions when the community experiences profound changes. Then, boundary decisions change the community boundaries by including and excluding individuals, objects, actions and rules. As a result, new governance practices emerge or existing practices are redefined, and the cycle goes on.

Figure 7: The Cycle of Governing OSS Communities

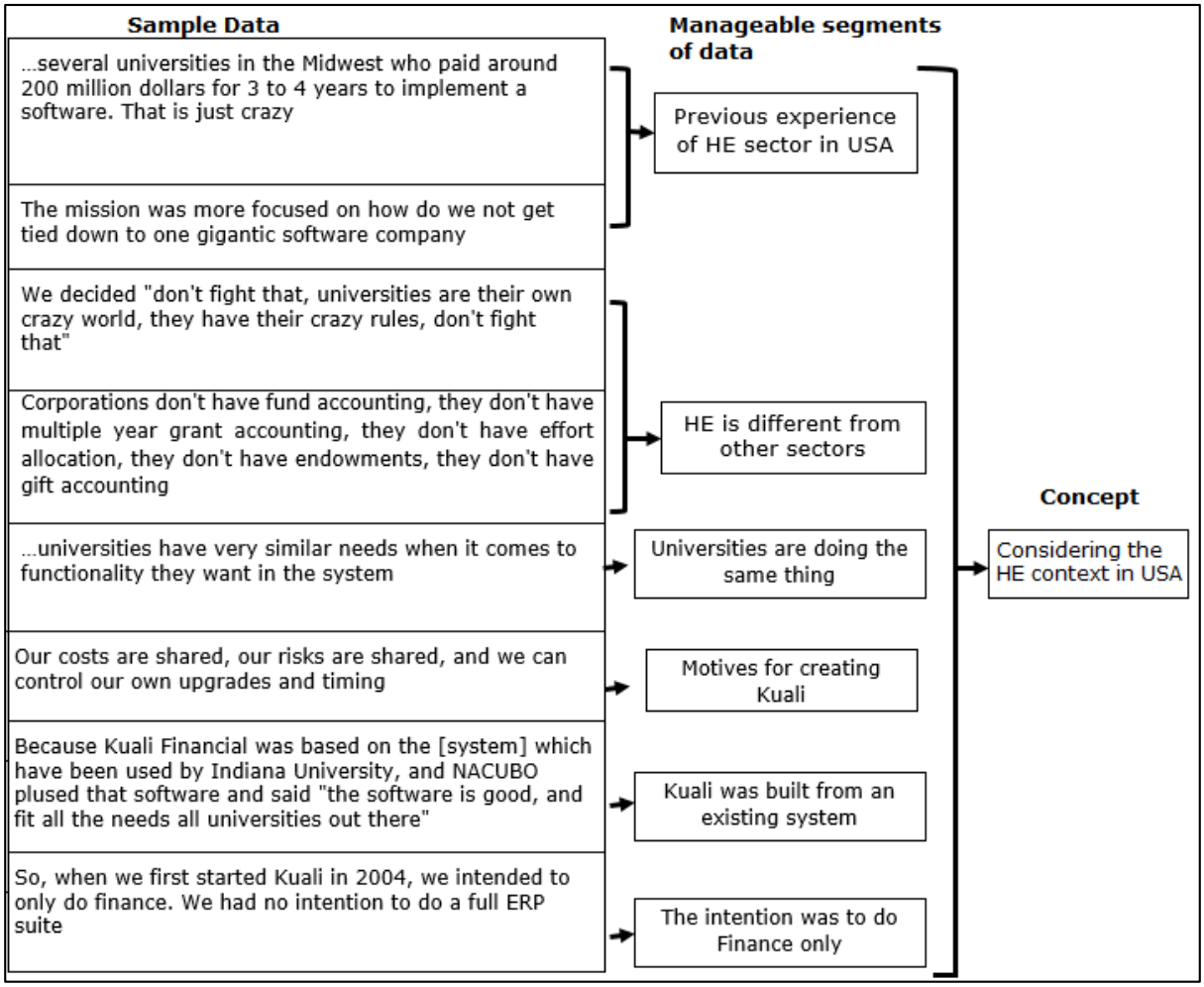


Source: Author's Own Figure

The processes of analysis and design were evident throughout Kuali life to gather the continuously changing requirements of the universities and set the community goals. Prior to the formation of Kuali community, the

processes of analysis and design were initially focused on understanding and considering the context of the HE in USA. It involved gathering and analysing the requirements of the potential stakeholders, and accordingly design the finance system, i.e. KFS. Figure 8 summarises the analytical process of moving from the research data to the processes of analysis and design.

Figure 8: Data Structure: Phase 1-Analysis & Design



Considering the context of HE in USA required understanding the previous experiences of the HE sector with regards to the development and procurement of ERP systems in general. Prior to the inception of Kuali, the American universities were mainly acquiring their ERP systems from giant software vendors, like Oracle and SAP (Admin1[8]). Besides their high cost,

[8] Please refer to table 8 in chapter 3 for a the list of interviewees

these systems were developed for the American market (S1[9]), and thus required a lot of customisations to meet the requirements of the universities. In addition, the software development, maintenance, and upgrades were bundled together preventing universities from controlling the software:

> "*…in those software companies, higher education is such a small segment of it, but* ***you have to upgrade because everything else is upgraded[10]****… Even if you don't need the functionality, you have to install it in order for this role to work*" (BA2)

S11, a speaker in one of the sessions of Kuali Days 2015, sarcastically recalled the struggle:

> "E*very implementation of a system involves a* ***full bucket of misery*** *and you have to eat it from the beginning to the end and you can eat it faster you can eat it slow but* ***you gotta eat the whole thing****. Those are* ***miserable experiences*** *and they cost us a lot of money, and a lot of time, and a lot of aggravation, a lot of coordination*"

Besides understanding the previous experiences, the analysis also involved understanding the specificities of the HE sector that has led to the paucity of HE-specific ERP systems in the market. The interviewees believed that universities are unique in comparison to other sectors, as summarised by BA2:

> "*We [i.e. universities] do fund accounting. Corporate world does not do fund accountings… Each fund has its own restriction, especially for research institutions… We also do budgeting. Budgeting is not important for corporate world. We do encumbrances, they don't...* ***[We] have a sort of a mix of a corporate… and higher ed in the***

[9] Please refer to table 7 in chapter 3 for the list of speakers
[10] The bold parts of the data illustrations are my emphasis to stress the meaning in relation to the concept being discussed.

***fund accounting**. So, the software has to be able to manage both without having to have issues*"

Despite their uniqueness, there were lots of commonalities among American universities in terms of the functional requirements of a finance system (Admin6). However, they continued spending money on software products that are similar, as explained in the following quotations:

"*…it just seems to me that it was **crazy for us to be spending this money essentially doing the same thing***" (Admin2)

"*…we all report to the same federal agencies. We have similar financial statement preparation… when we talk to people at other universities, all our faculty acts exactly like their faculty. And our purchasing department is very similar to their purchasing. So, **we are all coming from a very common ground***" (BA1)

As mentioned previously, analysing the requirements is essential for designing the intended software. In this context, design refers to the functionalities and the technical specifications of the KFS. The data revealed that the functionalities of the KFS was based on the existing finance application that was used by IU before forming Kuali community. According to S5, that functionality was approved by the National Association of College and University Business Officers (NACUBO); the finance director club in USA at that time. NACUBO approved the design and supported further developments because the functionality of IU's system was flexible to accommodate the needs of different types of universities in USA; however it was technically obsolete:

"*when they first started with the Kuali financial system, they took the Indiana's old legacy financial system, because **functionality-wise it was good**. So, they started with that*" (Admin3)

"*NACUBO, the national association of college and university business officers in the US, they were [essential players]. Because they were*

coming from the financial side of the world and they were looking at their member institutions spending 10 20 50 100 million dollars on the European software, not exactly the mission of the universities, and here is Kuali is offering an opportunity to spend a lot less money and get as much. They were very good supporters" (Admin2)

Wirth regards to the technical specifications, OSS was the chosen solution. However, before deciding to adopt the OSS approach, the founding universities initially thought of forming a HE-led for-profit company to implement the technical specifications of KFS in the conventional way and universities contribute to the development of the software. The idea was that this company will be led by the universities to avoid the exorbitant cost incurred by proprietary lock-in. However, the decision to take the OSS route is a means to an end for two main reasons. First, most of the universities believed that the proprietary route in commercialising software contradicts the ethos of universities in transmitting knowledge to the public. Second, it will create barriers for collaboration and code sharing. In other words, selecting the OSS option was not an intention. Admin4, a senior administrator in a university, recalls the discussions around this point:

"…one of the things that when we first started talking about Kuali, one of the things we began with is a discussion that we are going for a company, and the universities are going to contribute to it and make software and sell it and treat it like a product by this company, like a conventional way… ***most universities reflect in a very deepest way the notion that the purpose of which they fundamentally exist in transmitting knowledge from one generation to another argues strongly for openness****"*

The above quotation illustrates that the OSS path was chosen because it takes into consideration the values of universities, such as knowledge sharing. In addition, the OSS route realises the specificity of the HE sector and fulfils the common needs of the American universities, which were:

collaboration, cost reduction, and control (S7). This highlights the importance of context in creating and governing OSS communities.

The founding universities decided to pool their resources to build an open source KFS. They nominated functional and technical representatives to decide on the future direction of the KFS project. The intention was to create a university-specific, i.e. vertical, OSS community that has different settings than conventional OSS communities. This has raised concerns with regards to the boundary of the community:

> *"[O]n August 30 2004, Indiana University, the University of Hawaii, the NACUBO, and rSmart[11] put out an announcement that put IBM blush, that* ***we were going to build an open source financial system****… there is period of kind of figuring out* ***'ok, well, who is going to be in?'"*** (S2)

The emphasised parts of the above quotation illustrate the concerns regarding delineating the scope of the community, and the legitimate actors. This has triggered the first boundary decision; creating the community. This is further explained in the next section.

4.3. Governance Practice of Phase 1: Controlling the Code

Creating the community was a boundary decision that initially aimed to delineate the community borders to determine the objectives of the community, and who can legitimately contribute to it. It was also a boundary decision to include and exclude the material properties of the technology to meet the objectives of Kuali. Fulfilling the HE needs in USA was the main concern during that period of time, and thus controlling the code was given the highest priority to ensure a university-specific OSS. Figure 9 illustrates how "controlling the code" has emerged from the data. As shown in the figure, the code was controlled by fulfilling the HE needs, carefully choosing the OSS license, gating the community, and selecting

[11] A company that delivers IT solutions for the higher education sector

the resources. In this section, I will elaborate on each of these actions and routines.

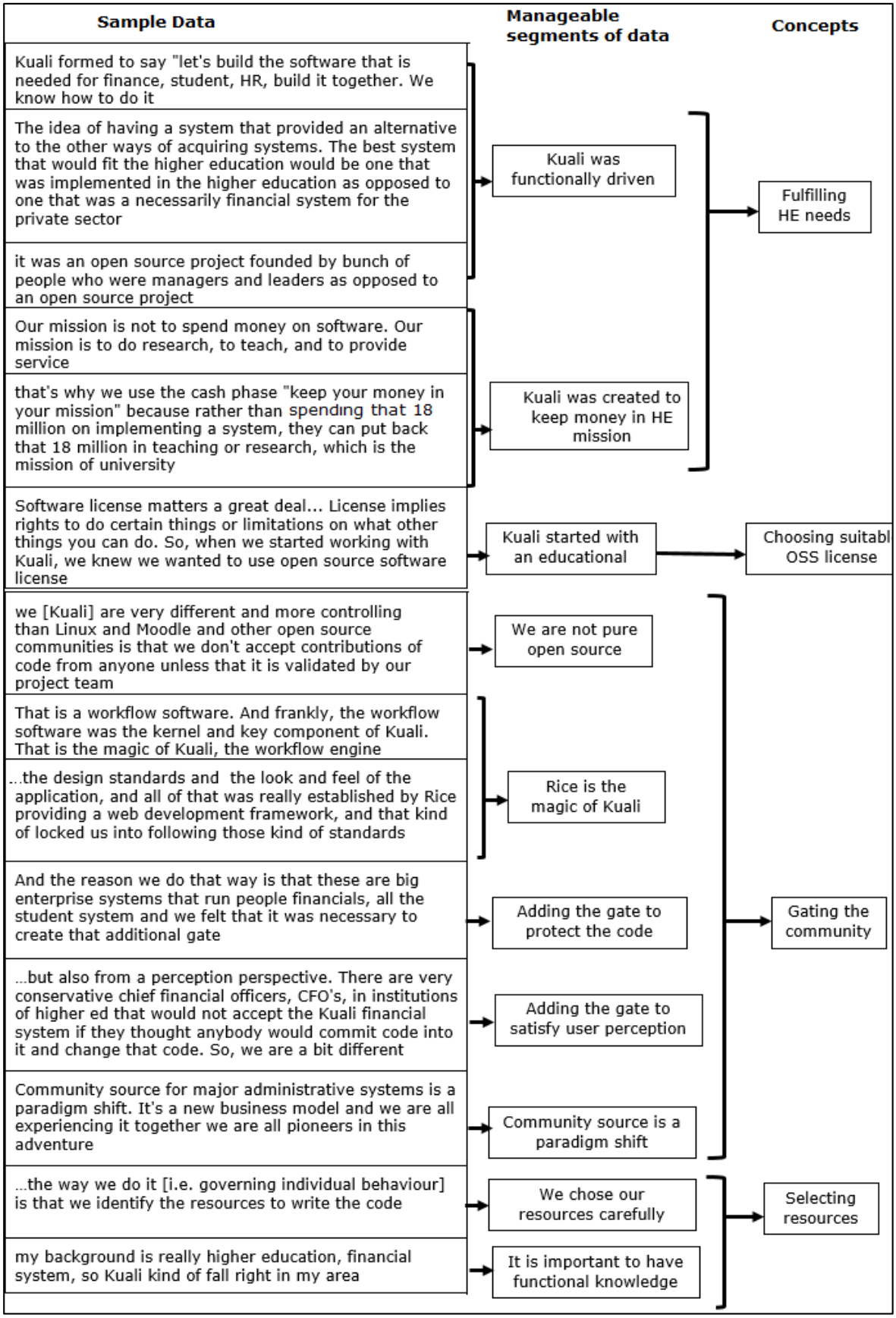
To fulfil the HE needs, the community founders had to explicitly define Kuali rules to distinguish Kuali from other conventional OSS communities, and from competing commercial ERP systems. The most essential rule was selecting the legitimate contributors of the community. The functional representatives, such as chief financial officers, were selected as the key players of the KFS project because they are the ones that best know the needs of the HE sector in USA. The functional representatives made sure to fulfil the basic needs at that time; functional value, and cost reduction:

> "*The big two motivating factors were **cost** and the fact that we thought **we could build a better system** than the vendors have built, more compatible with what higher ed needs*" (Admin1)

In terms of functional value, the role of the functional representatives was to maintain a functionally-driven KFS project:

> "*We basically said that the **rules are going to be defined by the functional owners, not the technical people**, **and this is another big difference**. Most open source activities are driven by the technical people. We said 'no no no, this is a software design for the business administrative support for universities, those people are going to make the decisions'* " (Admin2)

Figure 9: Data Structure: Phase1-Controlling the Code



In terms of cost reduction, the founders used the cash phrase "*Keep your money in your* mission" (Admin2) to attract universities to the newly developed Kuali community. Developing KFS was not free of cost; however they believed that

> "[*their*] *mission is not to spend money on software. [Their] mission is to do research, to teach, and to provide service*" (Admin2)

S5, summarises how implementing KFS was a cost-reduction project:

> "*There was no Kuali, those were the choices that we had. And there are some pretty eye-popping numbers.. so the quote was $23 million put that for financial… our [i.e. Kuali] implementations were in for less than $6m. So I can say to our Deans, we left at least $17m in your mission, for you to spend on research, for you to spend in scholarships, for classrooms, and such*"

The next important step to control the code was to choose the appropriate OSS license. The license enables and constrains the behaviour of community members, and thus facilitates controlling the code. According to Open Source Initiative, there are various approved OSS licenses that are used for different purposes. However, the community selected an Educational Community Source License; a non-restrictive OSS license, because it is aligned with the nature of the HE settings, as described by one of the founders:

> "*…we wanted to use open source software license… This software license says 'you can go and run with this stuff and do anything you want, you want to package it and sell it, you want to mix it with your stuff, you want to commercialise it'.* ***We really believed that a non-restrictive approach to reuse was the best thing in higher ed***" (S2)

After selecting the key players and choosing the license, the founders decided on the rules related to the software development process. This was

important because governing a pool of resources to produce a domain-specific OSS product raised uncertainty about the quality of the outcome. Therefore, the founders decided to gate the community. This means that Kuali community was not a pure OSS community; rather it was called a community source. The community source approach adheres to the same principles of open source; however it opens-up the OSS code, and gates the OSS development process:

> "*Anybody can download it [i.e. KFS]. But, the only people that were making changes are our product team itself*" (Tech3)

Admin1 explains the meaning of community source as follows:

> "*…there is a kind of demarcation line. So, anything that goes into the development and delivery of the software, up to that point, up to that line, is highly controlled to make sure that the software is at its highest quality. Once it is delivered and people started to download it, it is at the other side of the line, downloading it, using it and bringing it on to their campus, then they have complete control at that point on what they want to do with the software*"

Gating OSS communities has been previously discussed by de Laat (2007) who has focused on enterprise OSS products that are installed inside corporations and behind corporate firewalls, which limits the access to the insiders. However, this thesis presents a different type of gate that imposes restrictions using standard framework tools and workflow processes. Unlike conventional OSS communities that "*are pretty wide open in terms of development rights*" (Admin2), Kuali imposed a tighter control over the code development process by implementing Rice; a Kuali-specific workflow software.

Rice is the "*magic of Kuali*" (Admin2). It started off as a workflow engine that minimised and optimised the number of approval nodes within the processes of KFS. Then, it was further developed to include a middleware,

identity management, and a web development framework (PM1). Rice framework is a technical artefact that facilitated controlling the code of KFS. Rice also determined the design standards, and the look and feel of KFS. In fact, developers were locked into those standards, as summarised by Tech1:

> "*…the way the software is written, you follow the standards that were originally part of the software design**, and you do follow that because it is necessary to make the software work. The software is almost self-regulating** which is interesting… You have the initial design… It was constrained enough that **if you did not follow this design, the change you've made wouldn't work***"

Gating the development process using Rice framework is an example of how the material properties of the OSS code and the development artefact were modified to meet the objectives of Kuali. The gate was required to protect the code because the KFS project is a major system implementation. Some of the member universities have a multi-billion dollar endowments and budgets, and thus KFS was meant to be a significant enterprise system (Tech3). Admin2 explains the importance of protecting KFS at that time:

> "*…this software [i.e. KFS] is too important. With all the respect, Linux is a kernel of an operating system and it is very important, but… nobody is individually betting its institution's future on it, whereas our chief financial officers were basically going to be betting their financial systems and the financial information on this code, and we are not going to trust it to just about anybody*"

Adding the gate was also important to satisfy the perceptions of users. A gated community gives confidence to the university administration that KFS is developed in the same standards as the reputable commercial finance systems (Admin4). The potential community members are very

conservative chief financial officers in universities who would not trust KFS if the development is open to the public (Admin1, Admin2).

Besides gating the community, controlling the code was also implemented through carefully selecting the resources. During the first governance phase, resources were mainly focused on individuals and universities. Joining Kuali was a selective process because community source involves communication among known collaborators, as opposed to the conventional OSS communities. This means that joining Kuali was a selective process, not voluntarily-based. The selective process was applicable to both individuals and universities. The community founders selected universities that share the values of Kuali and have "*assertive respectful leaders*" (Admin2) that will assist in building the community. Admin2, describes the selection of universities as an assertive process. He recalls:

> "*There were many awkward conversations when I would go to universities to talk to them and they were thinking of joining and we would have the awkward conversation at the end of the day saying 'don't think this is the right fit'.' **We are not the right fit for you and you are not the right fit for us**'. In most cases, those were people I knew professionally but **I knew if I allowed the wrong kind of people in, it would not work**. So, we were pretty good about that*"

As explained in the organisational context of Kuali (chapter 3), Kuali community also included for-profit organisations, which were called Kuali Commercial Affiliates (KCA's). It was clear since the early days of Kuali that having a community of universities only would be a challenging matter. It was expected that universities will face various technical as well as administrative challenges associated with the development of the OSS code. Accordingly, it was crucial to include commercial affiliates as part of the Kuali ecosystem to provide paid technical and administrative services,

such as data migration, consultation, training, and support. However, including commercial affiliates to a not-for-profit open source project has raised doubts among the community members. Therefore, the commercial entities were selected based on their ability to share Kuali values and contribute to the public good:

> "*A few commercial partners wanted to become partners, and we basically said 'we don't think it is a good fit'.* ***We chose them very carefully***" (Admin2)

Kuali is a domain-specific community, and thus sharing Kuali values was not a sufficient criterion for selecting resources. Functional knowledge was a mandatory criterion as well while selecting functional and technical representatives of the contributing universities and KCA's. This concurs with the findings of Fitzgerald (2006), who has argued that experienced developers are required to ensure a successful development in vertical OSS communities. The following are excerpts from different interviews illustrating the background of Kuali representatives:

> "*my MBA is in investment banking. I mean I have a very good understanding of the world of finance*" (Admin2)

> "*my background is really higher education, financial system, so Kuali kind of fall right in my area*" (BA2)

> "*I've been doing this [i.e. developing ERP systems] for over 40 years, and I've been doing this for financial industries, manufacturing industries…*" (Tech1)

To sum up, during the first governance phase, Kuali community comprised of universities, for-profit organisations (e.g. KCA's), and not-for-profit organisations (e.g. NACUBO). Universities pooled their resources to build an open source KFS. KCA's are hired by universities to either contribute to the implementation of KFS or provide complementary services. Creating the community was a boundary decision to control the software

development process, form strategic relationships with KCA's, and target experts in the domain of HE. This was facilitated through implementing Rice; a restrictive development framework, and by selecting knowledgeable resources.

4.4. Consequences of Phase 1

The research findings revealed that the community source approach was a "*paradigm shift*" (S1) towards governing a vertical OSS community. The open source KFS and Rice framework enacted the practice of control and legitimised Kuali, not only as a software product, rather as a software development approach. Moreover, Kuali has defined new standards and best-practices that changed the HE settings in USA. The universities became the main controllers of their financial systems, where controlling the software was a desire among universities in USA.

Controlling the code provided a sense of community in the HE environment. As a result, more universities have joined the community because they were seeking for "*conformation that they are doing something the same way other schools are doing it*" (Tech1). Kuali became the trend in the development of financial systems among American universities. Consequently, other departments within American universities (i.e. other than finance) suggested expanding the scope of Kuali to develop a full ERP suite. This has triggered the need to redefine Kuali boundaries, and thus redefine the governance practices. This is further explained in chapter 5.

4.5. Conclusion

There are two main findings that are addressed in this chapter. First, this chapter illustrates the significance of the processes of the analysis and design in governing vertical OSS communities. I have showed how the context of the HE sector in USA played a significant role in setting the boundaries of the community and in the emergence of the main governance practice; controlling the code.

Second, the role of materiality of technology was evident in governing Kuali community during the first governance phase. In the first place, KFS was not built from scratch and did not follow the conventional OSS coding approach. Instead, the functionality of the KFS was based on IU's legacy finance system, which enforced developers to follow predefined specifications. Besides, Rice framework was designed as a Kuali-specific development artefact to control the software development process. The properties of Rice were set in relation to the context of Kuali during that time.

To sum up, creating the community was a boundary decision that identified the scope of Kuali community, which is building an open source finance system for universities in USA. Control emerged as the salient governance practice to achieve the objectives of the community. Creating the community was also a decision that determined the key players of the community, and identified the required resources. This boundary decision governed the collaborative effort of the community members until the founders and stakeholders decided to expand the scope of the community, and thus changed the objectives. This triggered the need for a new boundary decision, as will be further explained in the next chapter.

Chapter 5: Kuali Governance Phase 2: Balancing the Interests

5.1. Introduction

In the previous chapter, I illustrated how boundary decisions were essential to create Kuali community, set the objectives based on the context, define the suitable governance practices, determine the key players, and identify the resources. In this chapter, I demonstrate that boundary decisions are also essential in controlling and coordinating the large ecosystem, i.e. Kuali, while remaining sensitive to the changes that occur to the context.

After the success of KFS, Kuali community expanded its scope during 2005 from focusing on the finance system to the implementation of a full ERP system. Accordingly, the community became responsible for multiple OSS projects. Each project refers to a module of the ERP system. In this thesis, I focus on the research administration system (KC), and student system (KS), as well as the finance (KFS). Due to the expansion, Kuali invited representatives from different departments of the universities, and thus invited different interests. In addition, international universities were also invited to meet global HE needs, rather than restricting Kuali to the American context.

The objective of the expansion was to form a shared pool of diverse resources to enrich Kuali community. Kuali projects (i.e. KFS, KC, and KS) brought in new rules and resources, and reconfigured existing ones. In this chapter, resources refer to the attributes of individuals, capabilities, monetary resources, code, ideas, technology, and the divergent interests of the stakeholders. It was evident that each project utilised resources differently. Therefore, the expansion has caused profound changes to the settings of Kuali community. This has triggered the second boundary decision; balancing the interests, and accordingly Kuali entered the second governance phase.

This empirical chapter explains the second boundary decision; balancing the interests, and the governance practices during the second governance phase. The governance model during the first governance phase was based on controlling the code, which was insufficient to channel the resources among multiple projects after the expansion. Therefore, balancing the interests emerged as a boundary decision to aggregate and channel the resources in order to enable the growth of the community on one hand, and control the software development process on the other hand. This was facilitated by forming Kuali Foundation in 2006 as a not-for-profit organisation to govern the collective effort.

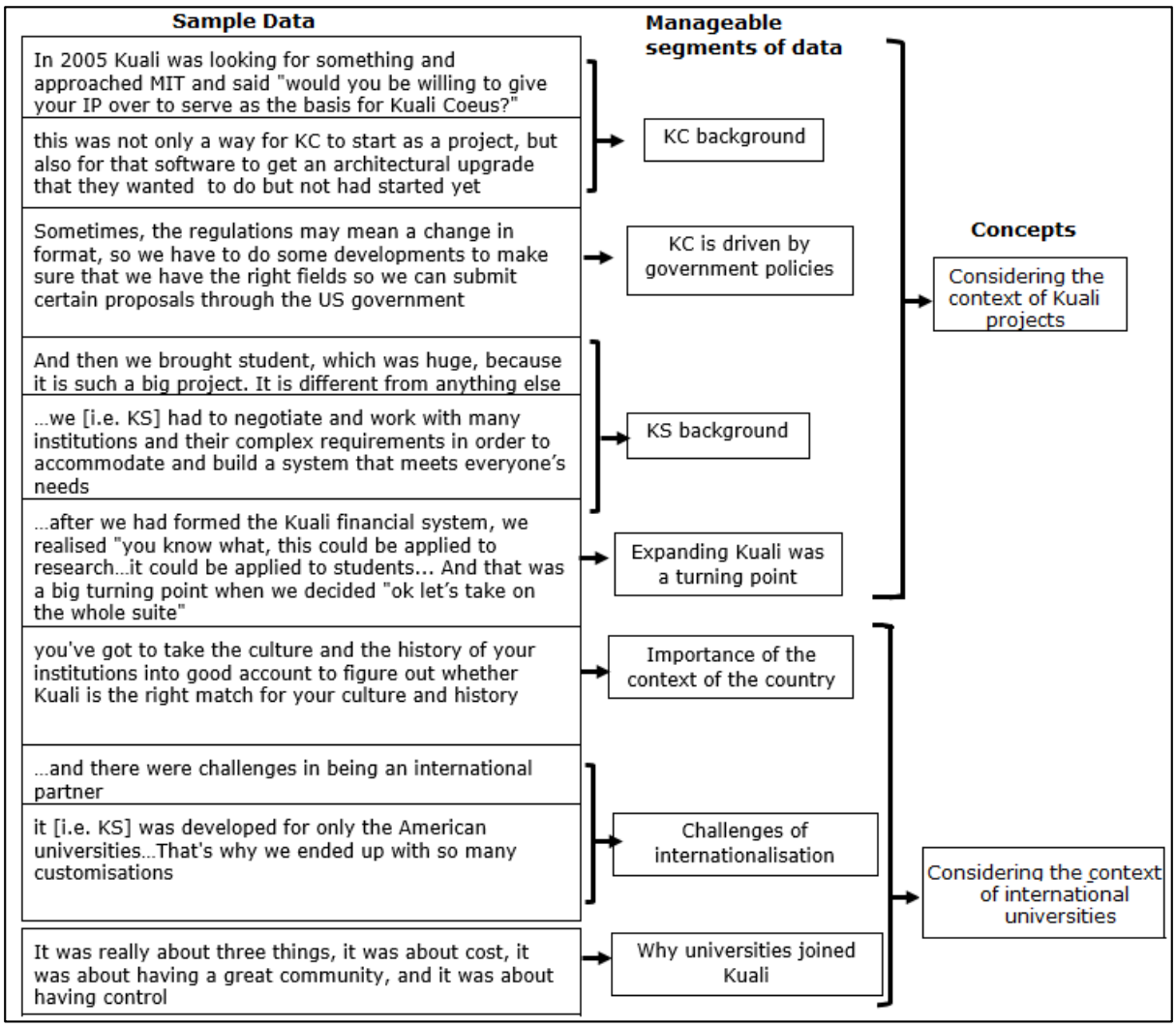
This empirical chapter represents the period between 2005 and 2014. The data illustrations in this chapter rely on multiple data sources in comparison to chapter 4. The illustrations are based on primary data (i.e. interviews) and secondary data, mainly email discussions, meeting minutes, videos, and documentations. Similar to the structure of chapter 4, I begin this chapter, in section 5.2, by explaining the processes of analysis and design, which refers to the conditions that triggered the second boundary decision. Then, in section 5.3, I explain how the collective effort was governed during the second governance phase. Then, I clarify the consequences of balancing the interests in section 5.4. Finally, the chapter is concluded in section 5.5.

5.2. Conditions of Phase 2

As mentioned in chapter 4, the processes of analysis and design are on-going in order to remain sensitive to the dynamicity of the community. Prior to the inception of Kuali, the processes of analysis and design were focused on what the American universities need. However, after the expansion of Kuali, the processes of analysis and design involved considering the context of each Kuali project, which includes analysing the requirements of the each project and its corresponding stakeholders. Besides, as Kuali invited non-American universities, analysis and design also focused on the international requirements of the HE sector.

Analysis and design were necessary to reset the community goals and identify the required resources for each project. Figure 10 summarises how the analysis and design of the second governance phase emerged from the research data. The following subsections explain the processes of analysis and design in details.

Figure 10: Data Structure: Phase 2-Analysis & Design



5.2.1. Considering the Context of Kuali Projects

The scope of Kuali community has been expanded to include multiple OSS. In the following lines, I provide a brief history of the newly developed systems; KC and KS, in order to better understand the contexts of Kuali projects.

Similar to KFS project, KC started-off from an existing system. It was based on the well-established non-open-source research system that was running at Massachusetts Institute of Technology (MIT), which was called Coeus. The Coeus project already had a community and a governance structure (BA1, Tech5). MIT internally developed Coeus 10 years before the inception of Kuali due to the lack of research administration systems in the software market at that time:

> "*…first of all higher ed is a small market. Oracle and SAP tried to panel their software for the higher ed. It wasn't designed for higher ed.* ***We are constantly trying to shovel a square peg in a round hole*** *when we are trying to implement some of those things… so when you start with a university it is already small market, and then* ***when you come down to 'who needs a research administration system', you've now narrowed the market even further***" (Admin3)

During the development of KFS in 2004, there were attempts to include basic functionalities for research administration; however they were limited:

> "*The KFS had some functionality for research; for preparing proposal budgets and managing grants when they come in, but that is all they had. We were still looking for solutions for things like regulatory compliance"* (Admin7)

In 2005, Kuali founders approached MIT and they have agreed on obtaining the intellectual property of Coeus system to be the basis for an open source research system, which was then named Kuali Coeus (KC). Kuali benefited from the available functionalities of MIT's Coeus system. At the same time, moving under the umbrella of Kuali enabled MIT to upgrade the technology through the collective effort of Kuali community. Therefore, the KC project was considered as a merger (S5).

Moreover, unlike KFS, the KC project was driven by governmental policies in USA, as described by Admin3:

> "*The rules around accounting are very clear in this country… And those rules don't change very often... In research, the rules change all of the time…* ***[O]ur rules [i.e. KC] change often and driven by this wonderful thing called Congress, and by the White House****… So, a lot of our policies are driven by 'what is the national institute of health is doing to us today?', 'what is NASA doing to us today*'" (Admin3)

On the other hand, the student system (i.e. KS) "*literally started from scratch*" (Tech6). Unlike KFS and KC, there was "*no history to build from*" (Admin6). Kuali partners refused to start from existing systems because

> *"…most of the universities that were partners on that project had mainframe systems and they felt they were like 50 years old. So, the approach that they followed was to say 'bring all your requirements and we will develop a new system'"* (Tech4)

The student system is huge in comparison to the other systems (Admin1, Admin6). In order to start from scratch, founding universities had to negotiate the complex requirements in order to accommodate and build a system that meets everyone's needs (Tech6).

Kuali project	**Background**
KFS	-Built from an existing system -Targeting finance staff (back office)
KC	-Built from an existing system -Already had an established community of universities -Influenced by governmental policies -Targeting different staff and students
KS	-Built from scratch -Complex requirements -Targeting different staff and students

Table 15: Background of Kuali Projects

As summarised in table 15, each Kuali project had different context, i.e. background, technical and functional requirements, stakeholders, and target users. Besides, each project "*brought new people into the game*" (Admin2), and thus new perspectives. There were misalignments between the newly developed Kuali projects and the existing settings of the community. This has triggered the need to shift the boundaries in order to accommodate the new requirements.

5.2.2. Considering the Context of International Universities

The processes of analysis and design were not limited to the context of Kuali projects. They also focused on the requirements of the international universities. As explained in chapter 4, during the first governance phase, the finance personnel in the contributing universities were the key players. Accordingly, the best-practices and the standards of Kuali community were based on the requirements of the American universities. However, inviting international universities entails considering the culture of the country and the history of the potential institutions (S6). Therefore, it was important to understand the requirements of the HE institutions worldwide.

When it comes to software development, the universities worldwide were seeking for cost reduction, collaboration with peers, risk sharing, effective technology, controlling the code, and compliance with best-practices:

> "*It [i.e. Kuali] is designed by us [i.e. universities], we hopefully have a lower cost of integration into our existing systems because the code is open, and we can unbundle support costs from the software. Our costs are shared, our risks are shared, and we can control our own upgrades and timing*" (S1)

However, there were variations in the university requirements within and outside USA, as summarised by S8:

> "*We've got the Hawaii system that has community colleges and big universities. We've got Toronto which is extraordinarily complex*

internally because they've got lots of different kinds of departments and they have both hierarchical and matrix governance structures. We've got South Africa which is it's the way South Africa does things completely different in many ways to the way they're done in the United States"

Besides, the research findings illustrate that each country had different motives in joining Kuali. In chapter 4, I have shown that the American universities were looking for functional value and cost reduction. However, the South African universities initially joined KFS because it offered a cost-saving exercise, regardless of the compatibility of functionality (Tech4). From a Canadian perspective, the open source path, not only provided an opportunity to collaborate with peer institutions, it also contained lots of escape routes in case they decided to progress on their own. Most importantly, Admin6, an administrator in a Canadian university, admits that US leadership was the main motive to join KFS project:

"*…from our perspective, our constant peer group… is heavily heavily dominated by US universities, private and public... Universities that we think of as peers, UC Berkley, University of Washington, these were folks that were in that community,* ***and hence it sort of legitimised it***"

In general, the variations in the contexts of Kuali projects and the international universities have consequences on the overall structure of the community and its objectives. Besides, the expansion of Kuali community involved the inclusion of resources that were not evident during the previous phase. Therefore, balancing the interest emerged as a boundary decision to govern Kuali community through balancing between community growth and control. This is further explained in the next section.

5.3. Governance Practices during Phase 2

Balancing the interest is a boundary decision that has emerged to start a new governance phase that recognises the specificity of each Kuali project in terms of its target users, software development approach, and investments. It also recognises the global requirements of the universities. Balancing the interests mainly focused on aggregating and distributing resources. In this context, resources refer to the attributes of individuals, capabilities, monetary resources, code, ideas, things, technology, and the divergent interests of the stakeholders. The expansion of Kuali community entailed different access to resources, and thus gave rise to divergent interests.

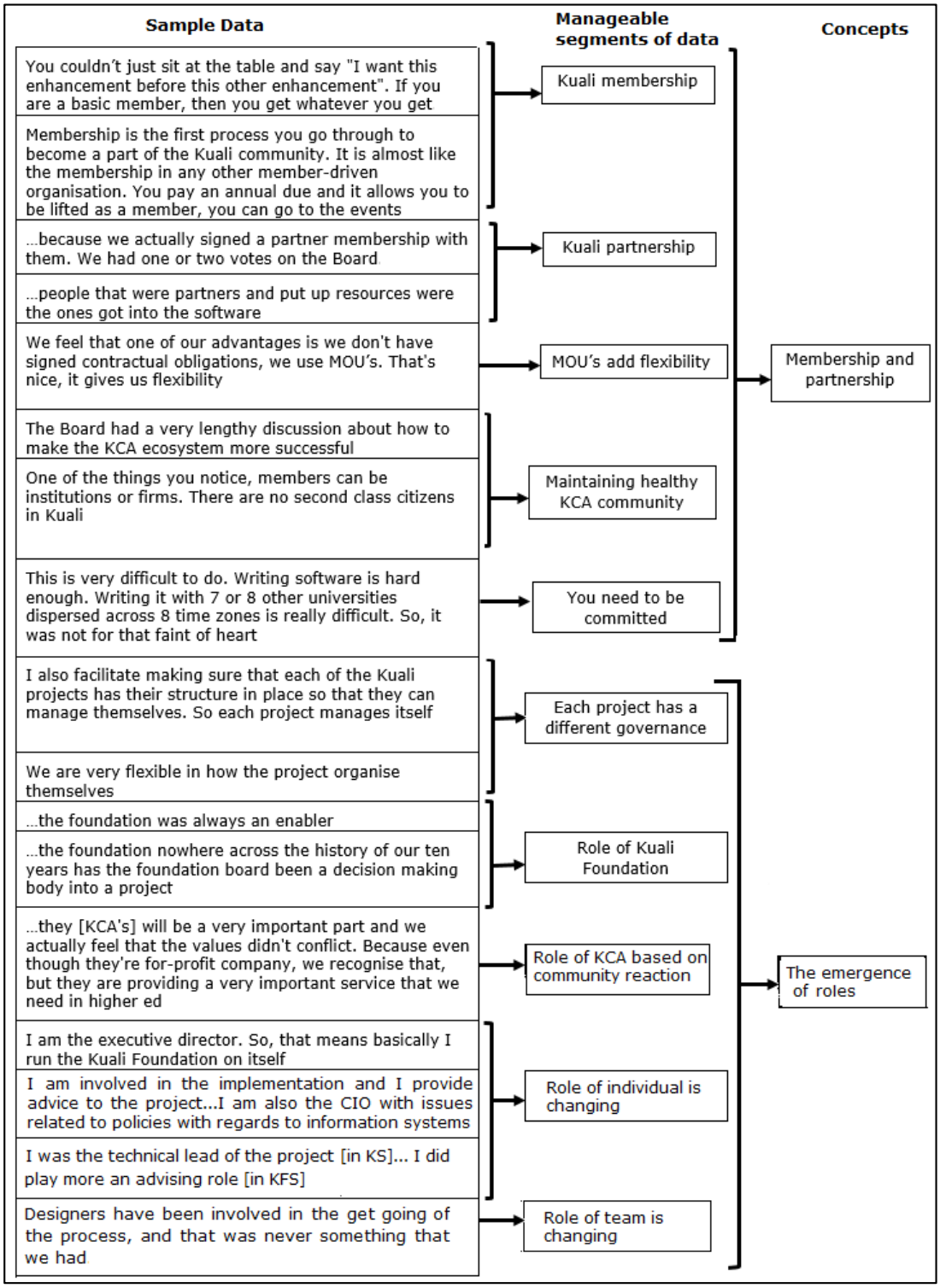
However, the existing governance mode at that time was solely based on controlling the code, which was insufficient to govern the growth of Kuali community and the emergence of multiple trajectories. Therefore, the community had to reconfigure the governance model to efficiently channel the pooled resources and coordinate the collaboration. During the analysis of research data, it was evident that the salient governance practices during the second governance phase were structuring the community, managing divergent interests, setting the properties of the OSS code, and facilitating collaborations. In the rest of this section, I explain each governance practice in details.

5.3.1. Structuring the Community

Structuring the community is a governance practice that defines the members of the community, their roles, and decision rights. This was facilitated through the formation of Kuali Foundation in 2006. The purpose of the Foundation was to enable collaboration among the distributed resources and reconcile conflicting interests. The Foundation comprises of board of directors and members that represent each Kuali project. The Foundation did not dictate how exactly each project should operate. Instead, it maintained a governance structure that allowed each project to

govern itself. As illustrated in figure 11, structuring the community was also achieved through setting membership and partnership processes, and through the emergence of roles.

Figure 11: Data Structure: Phase2-Structuring the Community



5.3.1.1. *Membership and Partnership*

The second governance phase focused on inviting diversity in order to enrich the community. However, the collective effort had to be governed through a membership process to regulate who is in the community and define the rights of the community members. Kuali membership required contributing an annual due based on the university budget to keep the community functioning:

> "*…if I speak in US terms… a big university like Indiana it's got about three billion dollar-a-year annual budget for eight campuses, you're going to pay twenty five thousand dollars a year just to be a member.* ***The software is free. You can get software without being a member, but just to make everything work*** *… For a very small college or school it might be five thousand dollars a year*" (S5)

Setting membership process was an important step to aggregate resources; however distributing the resources was done through partnership. Kuali partnership was essential to channel the pooled resources into different projects. When a university becomes a partner, it directs its resources towards a particular Kuali project. Kuali partnership was based on the golden rule; those who bring in the money, make the rules. Therefore, the partners who were bringing in functional and technical resources into the community in a full-time basis, and those who contribute higher annual fees, those partners were eligible to make the rules:

> "*When you become a partner in a project, based upon on that project funding structure, and they are all different, you pay an additional annual fee****… and that gives you a seat in the table for that project to outline priorities, to influence the project direction****... So, for that* ***the real value it is becoming a partner***" (Admin1)

In conventional OSS communities, rules are either implicit or scattered. In Kuali, rules were explicit and clearly defined in the form of memorandum of understanding:

> "*And our memorandum of understanding is a very simple document. Just a few pages that says how we're going to work together, that we agree that anything that comes out of this is an open source work*" (S10)

As mentioned previously, companies were also eligible to become Kuali partners, and thus the Foundation ensured to maintain a healthy community to avoid having "*second class citizens in Kuali*" (S2). The following is an excerpt from Meeting3[12]. It illustrates how Kuali treated universities and KCA's equally:

> "*-Members are members, and some are KCAs and some are schools; both are members of Kuali. Conflicts and perceptions could be applied to both.*
>
> *--**We want the best people regardless of KCA or school**.*
>
> *--If we see inappropriate behaviour, we should address it at the time, or by revising our principles. But, we should start out with equal status.*
>
> *--We should be more communicative in future when situations arise where this may occur. We want all KCAs to understand their options so there is not perception of favouritism*"

Either it is a university or a company, the Foundation needed a partner that made commitments, not just payed dues. To be committed is to have the

[12] During data analysis, each meeting summary was given a code. Refer to Appendix B for the meeting codes and their corresponding dates. Appendix B also includes some samples of meeting summaries.

will to commit and the capacity to collaborate (S7). The following is an excerpt from one of the regulation documents:

> *"The Partner Contribution Contract:*
>
> ***When you propose a contribution, you are agreeing to be responsible for that contribution until it is completed**. This is particularly important to understand for enhancement contributions.*
>
> *-Your functional staff must be available to write the functional specification, review with the sub-committee, and revise the specification as needed.*
>
> *-Your technical staff must be available to write the technical specification, review with the project team and technical committee, and revise as needed.*
>
> *-Your technical staff must be available to code the contribution, participate in code review(s), and revise as needed.*
>
> *-Your functional staff must be available to complete partner testing.*
>
> *-Your technical staff must be available to resolve issues found in partner testing AND in project QA."*

It is important to clarify here that the processes of membership and partnership were not limited to the process of identifying who is in and out of the community. Rather, these processes highlight that the properties of the universities were not fixed. Instead, they varied depending on their relation with the wider context, which has consequences on the decision-making rights. A university that pays the dues becomes a member, i.e. a regular user. A university that pays an extra due and nominate dedicated developers becomes a partner, gains access to change the code, and thus influence the project direction. Moreover, partners joined and left the community also based on their interactions with their surrounding context as emphasised in the below quotation:

"*over the 10 or 11 years, universities come and go, and it sometimes has to do with the* ***philosophical change****, sometime it has to do with the* ***change of leadership in the institution****, sometime it has to do with* ***financial constraint***" (Admin6)

The development of an open source ERP system, i.e. Kuali projects, was a complex collaborative activity that required multiple stakeholders to utilise various resources in order to communicate across university departments and beyond university boundaries. The employees in partner universities were assigned Kuali-related tasks besides their responsibilities within their universities causing blurred roles. Therefore delineating the roles was an important governance practice that has emerged after the expansion of the community.

5.3.1.2. The Emergence of Roles

Unlike conventional OSS communities where individuals self-select their tasks based on their interests and expertise, in Kuali, the roles and responsibilities of universities, companies, and individuals emerge from the negotiations among the different resources. The following explains the roles of the Foundation, the KCA's, and the partner universities.

Prior to the formation of Kuali Foundation, the community was running multiple OSS projects, where each had its own community and its particular approach to attract investors and code contributions. In other words, the community started off as a federation (S10). There was a pool of resources within each project (e.g. human resources, money, ideas), and a shared pool of resources for all (e.g. communication and development tools). The aim of forming Kuali Foundation was to channel these resources in order to build an integrated ERP system. Defining the roles of Kuali Foundation emerged from the desire to avoid any sort of exploitation of the collective effort:

> "*…the Kuali Foundation… is a legal entity. We have bylaws, we have elections for officers.* ***All of the members of the Kuali Foundation board serve as private citizens****. So, I am not the IU representative on the Kuali Foundation board of directors, I am [name] with my home address. And if I quit IU tomorrow I am still stayed at the board. So,* ***that means if I had a conflict of interest between what is good for the foundation and good for IU, my moral and legal responsibility as a director of the Foundation is the benefit of the Foundation****.* ***We made that very intentional in the design***" (S2)

The Foundation ensured consistent governance across the projects. Accordingly, all projects had similar governance structures, which comprised of: project boards, functional councils, project management, development teams, and subject matter experts. The Foundation recognised that each project had its own community (i.e. universities and KCA's), and thus enabled them to govern themselves. The Foundation never had been a decision-making body (S10). A senior administrator in an American university recalls:

> "*There was a discussion about what the role of the Foundation was going to be, relative to each of these projects.* ***The model that we picked recognised that each project had its own community****, and set of constituents who were going to make the cases with the institutions on their behalf to get investments….it wasn't the Foundation telling each of the projects exactly how to do what they had to do. We left the project boards, which represented the investing partners for projects, make those determinations*" (Admin4)

Kuali Foundation was responsible for providing generic shared services, such as managing Kuali Wiki and the bug tracking systems, packaging, branding, and marketing (S2). In addition, the Foundation was also

responsible for managing financials, controlling expenditures, and ensuring that the software code was open source license as intended.

In terms of KCA's, their main roles were related to implementation and consultation. Their roles were identified by the community reactions towards KCA's. There was scepticism towards having commercial entities in the community. At the same time, the community was aware of the large amount of additional technical and functional tasks that are associated with the development of OSS projects. Admin1, a senior administrator in Kuali Foundation recalls:

> "*…**there was an incredible scepticism about why in the world we would have commercial affiliates** alongside this open source non-profit foundation… Whether you buy it from a vendor, or you build it in your own, or whether you get an open source product like Kuali… you have a whole process of figuring out how to implement it… how to convert what you have…how to train people… how to integrate it… how to work with your data warehouse. All these different things and Kuali does not help you with that. So, many institutions want someone to help them. They don't want to do it all in their own. **They want help***"

The partner universities and KCA's dedicated their employees to work on Kuali projects in a full-time or part-time basis. Initially, the Foundation defined an explicit description of the roles and responsibilities of each individual in Kuali. In some cases, the number of expected working hours were also specified, as shown in table 16. The following are excerpts from KFS documentations illustrating sample roles and responsibilities:

> "***Project Team***
>
> *A functional lead will coordinate all functional and technical support and oversight for contributions with other project team members. The lead SMEs [Subject Matter Experts] for modules impacted by the*

contribution will provide functional approval for bug fixes; a functional lead will work with the functional sub-committees and functional council as needed to provide functional support, feedback, and approval for enhancements. A technical lead will provide technical support, feedback, and approval, for all issue types. The project manager will work with the project team and the functional council to provide approval for inclusion in the desired version." (Document Name: Partner Contribution Process)

"***Board Chair (1 PTE at 120 hours/year)***

Term Length: 2 years

-Lead board meetings, delegating to the vice chair as needed

-Meet with the program director prior to each meeting to prepare and develop the agenda

-Work with administrative assistant to track ongoing topics and action items

-Convene working groups as needed to make progress on specific action items

-Conduct board votes as needed to reach decisions on specific topics" (Document Name: Governance Roles)

Duty / Task	Annual Time Estimate
Weekly functional council calls and related prep (40 calls * 3 hours)	120 hours
Travel (see Annual Calendar - 1 week commitments)	80 hours
Attending board meetings and related prep (12 meetings * 3 hours)	40 hours
Miscellaneous planning (12 months * 10 hours)	120 hours

Table 16: Estimated Working Hours for a Functional Council (Source: KFS Documentation)

These explicit predefined roles were influenced by the legacy work practices in the HE sector. The findings revealed that different user groups negotiated their needs over time and across Kuali projects as necessary, and thus the

roles were emergent. For example, when Kuali first started, the functional representatives were the decision-makers. They were driving the future direction of KFS. The role of technical teams was limited to the design and coding based on the inputs of the functional teams. However, the expansion of the community altered these roles:

> "*Designers have been involved in the get going of the process**, and that was never something that we had***" (PM1)

The reason for that is the variations in the requirements of Kuali projects raised issues with regards to the design and the interface of the OSS. The community started to emphasise on user experiences, which were best understood by the technical teams.

Another example is the emergence of roles based on the context of the country and the university. Unlike the case in the American universities, the technical teams in one of the South African universities had a greater influence on Kuali projects from the very beginning. Tech4 explains:

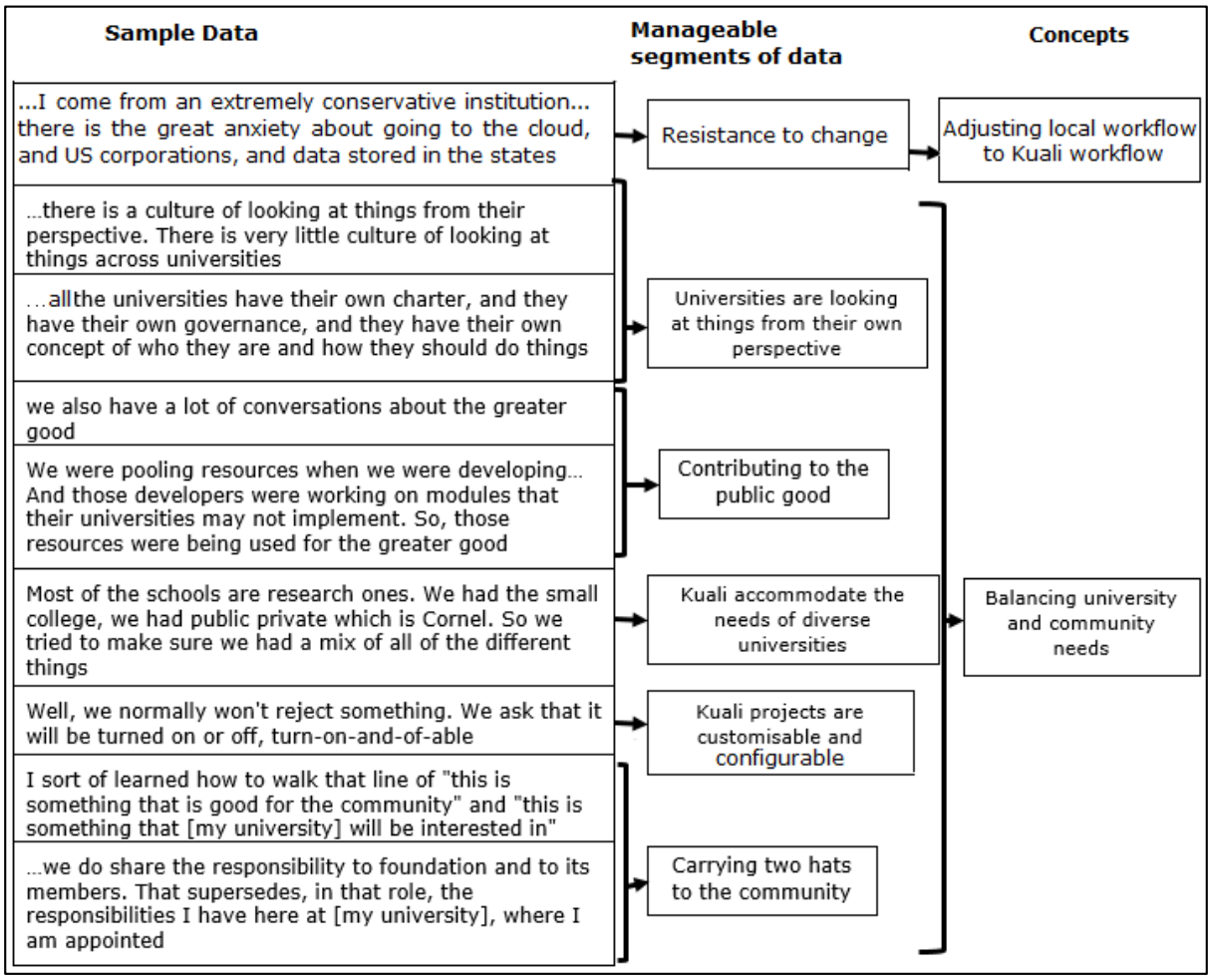
> "*One reason why that might be the case is before we actually implemented the software locally, some of our technical people were working on the project, helping developing software, but still for the American market… So, the technical team had more implementation skills specific to that particular software than the functional people.* ***So they had even more functional knowledge in some instances****. Obviously because they were more connected with the community"*

Although Kuali Board set fixed role descriptions, the dynamic nature of Kuali community resisted predefined roles. The above quotations illustrate that the roles and responsibilities were emergent. They were continuously negotiated based on the contexts of the project and country.

5.3.2. Managing Divergent Interests

Managing divergent interests was a governance practice that emerged due to the inclusion of diverse international universities ranging from small community schools to major research universities. As shown in figure 12, this required universities to adjust their local work flow processes to meet the needs of Kuali. Besides, the existence of diverse universities has raised the risk of the community being dominated by the requirements of leading universities. In other words, it has raised a tension between the university-specific needs and the community needs.

Figure 12: Data Structure: Phase2-Managing Divergent Interests



Kuali community include universities that are entrenched in their work practices (Tech1). Universities have divergent interests, and thus the community experienced resistance to adjusting university-related work

practices to meet the standard working practices of Kuali projects. One of the major resistance was with regards to the process of purchasing OSS in American universities, as explained by S5:

> "*…one of the things we're working on this year [2014] is working with our procurement or our purchasing agents.* ***We're helping them understand how to buy free software, and this has been a long journey***"

Universities are used to purchase software through a tender; however Kuali Foundation is not a company. This complicates the process of joining Kuali community. Managing divergent interests is a governance practice that looks at such complexities and adjusts the work practices from both perspectives. The following is an excerpt from meeting2, where the Board members were discussing solutions for the tedious workflow process of purchasing an OSS product:

> "*[A member] shared that the [a company] is interested in providing an overall marketing and outreach program for a variety of open source higher ed projects, including Jasig, Sakai, Durapace, and Kuali. This project is a start, to show value and to create a deliverable that is useful. This project will help in analysing the procurement process and how we can make open source software acquisition more viable in the process*"

In addition, the diverse universities of Kuali community have emergent university-specific requirements. These requirements were mainly implemented through two different ways. First, the university performs the implementation internally using their local resources. Then, the university proposes to contribute this newly added feature to the base-code (i.e. common pool) by going through an agreed-upon approval process. Second, if the university lacks the resources to implement the feature locally, it requests the corresponding Project Board to discuss the possibility of implementing the feature using the pooled resources of the project. In this

case, this specific feature will not be implemented unless it is aligned with the common community needs. This has caused a tension between the community requirements and the university-specific requirements, especially that universities are looking at things from their own perspective. They view Kuali:

> "*1) as an ERP vendor that provides systems they can install, or 2) as a community that they can engage with that provides a strategic way to move their administrative systems to a new, higher quality, and more cost-effective approach*" (Meeting28)

However, at that time, Kuali community was not a vendor. It was a community-source:

> "*…it [i.e. Kuali] isn't a zero-sum game. This is a game in which winning is possible across the board. But it may mean that no one school gets exactly what they want all the time*" (Admin4)

Project Boards accommodated this tension by inviting diverse universities to be partners in Kuali in order to ensure the coverage of wide range of requirements. With regards to KFS and KC, the system requirements were clear, and thus the focus was on attracting variety of institutions:

> "*For instance, financial systems have 12 partners, all the way from very small schools including community colleges, up to major research universities. Kuali Coeus, the research, has, I think, 18 partners now, again very diverse from small and large*" (Admin1)

> "*…we tried to make sure we had a mix of all of the different things*" (BA1)

However, with regards to KS, the requirements were complex. Thus, diversity was not sufficient. The KS community was targeting universities that have complex business requirements in order to share their experiences:

> "*We wanted to have rules for complex relationships… We wanted to identify complex business requirements we saw common across many different large universities. We wanted to bring that, those features, out of the box"* (Tech6)

The tension between university and community needs was also reconciled through restricting the process of code contributions. This was achieved by either parameterising or prioritising contributions, which highlights the materiality of technology. Parameterising features takes into consideration that the newly added features may be relevant for certain universities, but irrelevant for others. In addition, the relevance of the feature may change over time:

> "*When I understood it was something that was very specific to [my university], I made sure there were feature flags, parameters**, you can turn this feature off or on**. I did the extra work in the software to accommodate that*" (Admin3)

> "*if these things are going to be sharable in the future, I will keep that in mind when I write the functional spec. For instance, Hawaii might use this, Cornell might use this, so even though California only needs this, I parameterise it*" (BA2)

In terms of prioritisation, some contributions were given higher priority due to their higher impact on the community. For example, Admin7 describes prioritising features in KC project as follows:

> "*But I think a lot of us feel like the proposal development module being something that a lot of people can use is a greater good issue. It also helps us to get new users, new partners to the community, whereas a small module who is only used by central administrators at a university may not be the thing that gets us new partners*"(Admin7)

It is evident that parameterising and prioritising the code contributions is a way to arrange the material features of the OSS code with relation to the context. Managing divergent interests shows that the OSS code in not only an end product. Instead the code is co-developed and in a continuous state of change. The material features of the code are rearranged to enable the accomplishment of the governance practice. Further illustrations are provided in section 5.3.3.

The community also reconciled the tension by allowing the emergence of roles. Same individuals were sometimes assigned different roles causing different interactions, and thus different consequences. For example, some Board members toggle between two roles: a university representative, and a community representative. During the Board meetings, these roles emerge as a significant agent that determines the direction of the project (i.e. whether to accept the proposed features or not):

> "*So **I sort of carry two hats** though. I need to carry my hat of 'this is what [my university] needs', and then I have to have a hat that says 'this is what is good for the community'. My boss understands that sometimes I am conflicted in those two roles*" (Admin3)

> "*…when you are an institutional member and you are voting on the road map items, you are paying in your funds to be a paying member, and you are definitely thinking about 'well what am I getting from this money?' or 'what I need might be different from what my companion school who is sitting next to me might need'… So, we have to balance 'what do I really need' versus 'what has to get the community going'*" (Admin7)

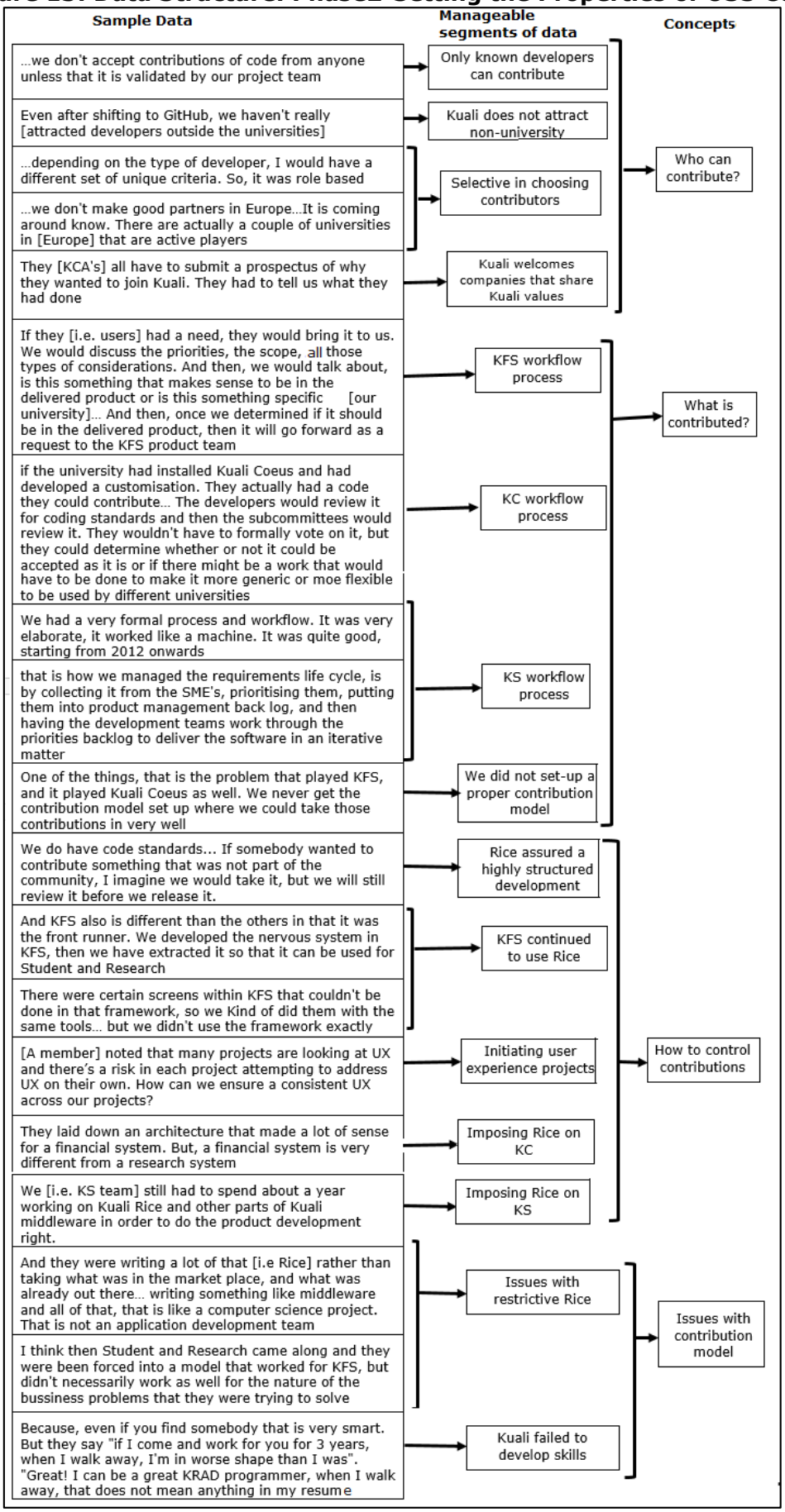
In OSS communities in general, tensions cannot be totally eliminated. The fluid and dynamic nature of the community promotes the development of diverse requirements and interests. Therefore, managing divergent interests is a governance practice that realises the importance of fluidity; i.e. existence of divergent interests to enrich the community.

5.3.3. Setting the Properties of the OSS Code

The collaborative effort of Kuali members is directed towards the development of software code. This code is shared and continuously modified to cope with the dynamic context of Kuali. Therefore, the code is not only a product, it is a resource that is co-developed through the interactions. The OSS code in Kuali community either refers to the base-code or to the contributed code. The former is the code of the shared OSS projects that is co-developed and used by the distributed teams. The latter is the code that is often developed locally within a particular university as a result of an internal requirement (i.e. university-specific). The university then proposes to contribute this newly developed code to the base-code in order to be accessible and used by the community.

Kuali Board members agreed on setting restrictions on the process of code contributions. These restrictions are referred to in Kuali documents as the "contribution model" (Meeting4). The OSS code is considered as a resource, and hence the "contribution model" sets the standard properties of this resource. The properties of the OSS code determine who can contribute a code, what is contributed, and how to control the contribution, as summarised in figure 13. This section illustrates how the properties of the OSS code were arranged and rearranged in relation to the context in order to control the code development process.

Figure 13: Data Structure: Phase2-Setting the Properties of OSS Code



5.3.3.1. Who Can Contribute?

> "*…the only people who could develop the software were the **known developers working for the universities**… if we went and had a thousand universities all telling us what we should do, we would never get anything done*" (Admin2)

The employees of member universities and KCA's are the only eligible contributors of the community. This property has been determined by the specificity of Kuali projects:

> "*Kuali is a huge ERP system. It **is specifically focused to the higher education** community. It is pretty **difficult to sort of understand and implement** and it usually takes a lot of time and effort to understand and implement. So, I mean that sort of **limits the interest to get out of the community***" (Tech3)

The above quotation clarifies that Kuali did not attract the interests of outsiders, nevertheless joining the community continued to be a selective process during the second governance phase to control the code contribution process. However, due to the distribution of teams across projects and countries, the selective process was reconfigured. Unlike the selection criteria during the previous phase that were limited to functional knowledge and sharing Kuali values, choosing team members during the second governance took personal skills into consideration (Tech6). Functional knowledge alone became insufficient as the developers started to collaborate with wider range of resources. For example, in a highly distributed team, one of the main criterion is the individual's ability to work with physical and virtual teams:

> "*…for people that are actually going to be working in the product team, we also were considering how well they will be able to work with other people on the team and particularly their ability to work in*

a remote team because all of our teams were remotely distributed" (Tech3)

The process of selecting universities was also reconfigured, especially that the community was aiming for internationalisation. When Kuali first started, the founders avoided the inclusion of European universities because

> "*they were way too parliamentary. They were too didactic in their requirements, you know, 'this is the only way will work for us', and [the founders] gave up*" (Admin2)

One possible reason was that the community at that time was looking for universities that were flexible to accommodate the needs of global HE sector. On the other hand, universities outside USA were uncertain about the outcomes of Kuali. However, gradually, partners from Europe joined in when they have realised that Kuali projects are capable of handling international requirements.

The specificity and the complexity of Kuali projects limited the contribution to the employees of the partner universities and KCA's. Through time, the Foundation realised that they have failed to reach other potential contributors who can enrich the base-code. Therefore, the Kuali Board members decided to approach potential contributors during Kuali Days; especially universities that were joining the event to know more about Kuali, i.e. "tire kickers":

> *"[A member] was intrigued by those [i.e. schools] that are interested in engaging but are struggling to find the right avenue to begin. [A Board member] specifically said that while we are all at Kuali Days, it's important to focus on the conference and the people attending rather than our internal meetings"* (Meeting28)

This suggests that "who can contribute" is an emergent property of the OSS code. It is defined and redefined depending on the context.

5.3.3.2. What is Contributed?

"What is contributed" is another essential property of the OSS code. It includes the standard criteria for the contributed code. First of all, the contributed code had to be generic; useful to the community as a whole. It had to be customisable and configurable to ensure flexibility and reusability. This was also essential to manage the divergent interest, as mentioned in section 5.3.2. Each Kuali project had its own Board in Kuali Foundation. The contributions were initially assessed within the requesting university before they were officially proposed to the Project Board. The process of approving a contribution varied depending on the regulations of the university and the corresponding Board.

In USA, the workflow process was typical with regards to KFS project (PM1). BA1, a business analyst in KualiCo and a former functional member of KFS team in Uni3[13], describes the KFS workflow within Uni3 and how it was approved by the Board:

> "*Within the university, a business need is identified. And, then that is prioritised with all the other things that the people are asking for. And then, it is developed, tested and implemented at the [Uni3]. And then, when the [Uni3] is ready to contribute back, they submit their proposal to the customer advisory group [previously known as a functional council]. Then the advisory group reviews that, and maybe add additional requirement, or indicates something that needs to be parametrised, so that it can be turned on or off, because maybe some schools don't want it exactly the way [Uni3] has developed it. And then, once it has been approved by the advisory group it goes to the product team [in the Foundation] for review. And then the product team pluses it, and then [Uni3] would develop it and test it. And then, it would come into base*"

[13] Pseudonyms are used to preserve anonymity and confidentiality.

With regards to KC project, governmental policies were part of the approval process, and thus KC team adopted a different workflow process to assess code contributions:

> "*The first thing we are looking at is are any of the regulations around research changing and do we need the software to accommodate those changes. Sometimes, the regulations may mean a change in format, so we have to do some developments to make sure that we have the right fields so we can submit certain proposals through the US government*" (Admin7)

The workflow process of KS project was more complicated. As mentioned in section 5.2.1, KS started from scratch. It did not have any basis to start from, and thus the development team was more focused on agreeing on a suitable architecture for the KS (Tech6). Accordingly, the workflow process started to stabilise by 2011/2012. At that time, KS teams started their implementation using agile methodology[14], and accordingly had a different workflow process in comparison to the other Kuali projects:

> "*…when new functionality was identified, we typically wrote epics and user stories[15]. We followed the agile methodology to actually get them developed and rolled out. Because the Student project was still under development, there wasn't a direct impact to the business because none of the schools or universities that were partners that actually putting that directly into production*" (Tech4)

> "*So, in fact, we would take the high priority requirements, put them into agile development methodology that the team to work on to rollout every 2 weeks into a working product. Then every 2 to 3 months we would cut the milestone, a milestone would be a complete functionality, a set of features that anybody could take and use. And*

[14] Incremental and iterative software development methodology.
[15] Epics in agile methodology are big user stories with broad scope. They have to be broken down into smaller stories before the team works on them.

then multiple milestones would roll up, let's say, over a period of a year, we will make a release" (Tech6)

The agile methodology involved collecting the requirements from the university representatives, prioritising them, and assigning them to the product management to start development. The developers would then communicate on daily basis with the university representatives to verify and validate the work. Then the software is delivered in an iterative manner. This suggests that each project team within the partner universities have their own way of processing contributions internally, and then contributions were processed in the Foundation in order to be added in the corresponding base-code.

As previously mentioned, the contributed code had to be generic and customisable in order to be added to the base-code. However, the process of approving the code was tedious. This has encouraged universities to fork Kuali projects. This means that, by allowing excessive flexibility and imposing tight rules, some universities downloaded the code and continued working on it in isolation; without contributing back their work, to avoid going through the tedious approval processes. In other words, universities kept their contributions locally instead of expanding and improving the base-code. This is similar to implementing an in-house application and it contradicts with the main purpose of forming a collaborative community. Thus, Kuali became inefficient (S1):

"*The process to get a contribution accepted back in the base-code is difficult and is something we need to fix because we definitely want to know what people are doing, what enhancements, and fixes they are making that we can incorporate and not have to offer*" (Admin7)

Accordingly, the Foundation responded to this issue by resetting "what is contributed". The Board members revised the current contribution criteria and suggested alternatives. For example:

> "*The guiding principles are to 1)* ***keep the bar low for initial contributions, so people don't begin a fork with other places*** *for this; 2) to have the bar high for giving our QA seal of approval; 3) to gain community support; and 4) to engage at initial conception so people don't wait until they have something already complete… [S]haring up front builds collaboration and ensures future QA acceptance.*" (Meeting 4)

Resetting "what is contributed" encouraged the community members to collaborate and share their contributions. This was essential in order to expand the base-code and enrich the community, which was the main objective of the second governance phase.

5.3.3.3. How to Control the Contributions?

The property "how to control contributions" focused on designing the framework that regulates the OSS code according to Kuali standards and objectives. During the first governance phase, contributions evolved around the KFS project. Controlling the contributions at that time was achieved through the implementation of Kuali Rice. This was considered, from the community's perspective, as the best-practice that maintains a vertical OSS community. Therefore, Kuali Board members agreed to impose this best-practice, i.e. Rice, on all other Kuali projects after the expansion.

However, during the second governance phase, each Kuali project formed an OSS community that develops an ERP module. By imposing Rice framework, the Board members believed that Rice will channel the resources to the different projects, and will control the software development process. In other words, they presupposed that Rice will facilitate controlling the code. However, the data illustrates that there are no best-practices in OSS development. Instead, the practices were continuously negotiated in Kuali community. In the rest of this sub-section, I provide an overview of Rice project. Then, I illustrate how different Kuali projects responded to the act of imposing Rice framework.

Rice was an OSS project that was funded by all other Kuali projects. It consisted of a separate development team that was responsible for gathering the requirements of the various Kuali projects. Rice framework was imposed on all Kuali projects due to the nature of the HE settings. It has assured a highly structured development process (Admin1, Admin2). Therefore, Kuali community did not tolerate developers who had their own ways of doing things:

> "*All those cowboys and cowgirls, as I would call them, quickly washed out of the game. Did we lose the possibility of really talented people? Sure. But, we're not writing rocket science software. We are writing business software. This is not that complicated. And so, anybody who had really good skills in Java, we knew they could do it*" (Admin2)

Rice was initially created to act as a workflow engine for the KFS project. Then, it has been extracted and imposed on the other Kuali modules. The material properties of Rice had different implications on each Kuali project, and thus project teams responded differently. The following is a summary of the consequences of imposing Rice on Kuali projects.

KFS was the frontrunner, and thus Rice framework was developed in KFS. However, Rice was initially designed with the data-entry user in mind; the type of end user who does not care much about the design of the interface. This type of design neglected the fact that Kuali became an integrated ERP system that includes different types of users, and thus the appearance of the system became a priority as explained by PM1:

> "*…it [i.e. Rice] has been a* ***blessing and a curse****. There've been a lot of benefits, like getting that standardisation, getting that common look and feel across the products...* ***The huge drawback is it's really really limiting in terms of what you can do with user experience****"*

Accordingly, KFS teams started to deviate from Kuali Rice in designing their own user interfaces. PM1 continues:

> "N*ow we are moving away from more kind of, starting to pull back from the Rice web development framework, we just call it the Kuali nervous system, and* ***looking at using style guides and design principles****, and things like that to guide the common look and feel****, rather than being locked*** *into a proprietary web development framework that presents a lot of inflexibility in terms of the design*"

However, deviating from Rice framework raised concerns about the reliability of the alternative design tools. Consequently, the Foundation initiated a project to look into the user experience issues across all Kuali projects. The aim of the user experience project, or UX as known among Kuali Board members, was to enhance the interactions between the users and the systems:

> "*[A Board member] noted that many projects are looking at UX and there's a risk in each project attempting to address UX on their own. How can we ensure a consistent UX across our projects?... [She] also noted some schools are hesitant to adopt our products despite functionalities due to inadequate UX. She proposes a 1-year project to form a UX team with designers and developers to work with individual project teams*" (Meeting29)

With regards to KC, it was a well-established community, as explained in section 5.2.1. Accordingly, it reacted to the restrictiveness of Rice in a more proactive way. Instead of deviating from Rice, it utilised its well-established community and experiences in restructuring Rice framework. However, KC project team went through a series of costly projects. Admin3 explains:

> "W*e got the Kuali nervous system, the KNS... Then, Kuali decided that KNS is getting old, and they have used OJB [i.e. Object/Relational mapping tool], and they decided that they should*

go to JPA [a Java program]… At the same time they said 'we need to get off KNS and go to KRAD, the Kuali Rapid Application Development'… the board wanted the projects to go… So, in Kuali Coeus, ***we decided to go by the wishes of the board and jumped in first*** *and went with KRAD. We took one of our major major modules and converted it from OJB to JPA, and converted it from KNS to KRAD.* ***And it costed us few million dollars to do that***"

Similarly, KS teams suffered from the restrictiveness of Rice. Although Rice provided useful infrastructural services for KS, it has imposed an outdated framework that did not meet the requirements of KS stakeholders. The developers had to spend 2 to 3 years working on Kuali Rice to customise it. Tech6 explains:

"…we were also ***straddled with a legacy Kuali Rice*** *which was not the right middleware to move forward with for the next generation student system. So, Kuali* ***Rice was a drag anchor*** *that caused our development velocity to be less than half of what is should have been. If my team had the ability and the choice to use a new framework… we would have been developing at 2 or 3 times the velocity*"

One of the teams in South Africa proposed a different framework; however they have failed to integrate it with KS. Tech4 recalls:

"*It [i.e. Rice] was constraining…. **So, Student decided to use [a different framework],** which was not at all a Kuali framework. So, we had to board-out lots of stuff, and they [i.e. Foundation] saw that it's not working. Then Rice started to make changes... **we were forced to use that.** The Student project could not say they are going to use their own thing again because they did, and it didn't work either. So, we were forced to use that.* ***But, that brought a lot of complexity***"

Accordingly, KS partners lost their patience. Tech4 continues:

"*[the development] just took too long. And that's why some partners actually left the project as well because it took too long to get to the point where they get functional value out of what was developed*"

The restrictiveness of Kuali raised significant architectural issues. Upgrading Rice required upgrading all projects that are deploying Rice, which is costly to sustain (Meeting5). Rice was critical to the success of all Kuali projects, and thus Board members suggested to decouple Rice from the projects:

"***Rice is facing a perfect storm:*** *its expenses exceed its income… and our expectation for the project is that it does more, rather than less. There seems to be a disconnect between Rice's governance and its dual identity as middleware and shared services and its income path… The proposal is to separate Rice development from shared services and to fund each component differently"* (Meeting29)

Besides the misalignments between Rice and the Kuali projects, Rice also hindered the development of the technical skills within the community. Kuali Rice was programmed using Kuali-specific programming language called KRAD, which had consequences on recruiting developers. Therefore, there was a high dependency on core developers in each project (Meeting5). Admin3 explains the recruitment issue with regards to the American universities:

"*but the problem with that… put an ad in the newspaper and say 'we need a programmer that knows KRAD', guess how many resumes I am going to get? [laugh] I can't get any… even if you find somebody that is very smart. But the say '****if I come and work for you for 3 years, when I walk away, I'm in worse shape than I was****'. 'Great! I can be a great KRAD programmer, when I walk away, that does not mean anything in my resume'*"

Recruitment was more difficult for international universities, as described by Tech4, a member of KS team in South Africa:

> "*…because we were the only university using the Kuali product [in South Africa], you can't just go out the street and find somebody that knows the framework… That is the thing you don't want to be in open source. You want somebody from the street that will be able to actually maintain it and look after it"*

The problem of building skills and talents continued to be an issue throughout the second governance phase. This was evident in the Board meetings during 2013. For example:

> "*we aren't building up our knowledgebase*
>
> *--we have gaps when schools pull their people back*
>
> *--are we doing enough to develop new talent?*
>
> *--are we doing enough to sustain existing talent, including institutional talent"* (Meeting 39)

The second governance phase focused on the possession and distribution of resources. However, the Foundation neglected the fact that Rice was a resource that had to be bounded:

> "*I think we made a mistake that made Rice a bigger body of code, a bigger body of services, and therefore instead of being the essentials of messaging and identity management, organisation structure and definitions and things like that… it became a repository for other code pieces that has many applications… It became harder to keep going forward building new functionality dealing with the legacy of code that has been out in Rice*" (Admin4)

The research findings revealed that the restrictiveness of Rice has been entangled with the community choice to control the code development. In other words, coupling Rice with all other OSS modules of the ERP (i.e. KFS, KC, KS), and designing Rice using a Kuali-specific programing language

illustrate how the material properties of technology are interrelated to control the OSS code.

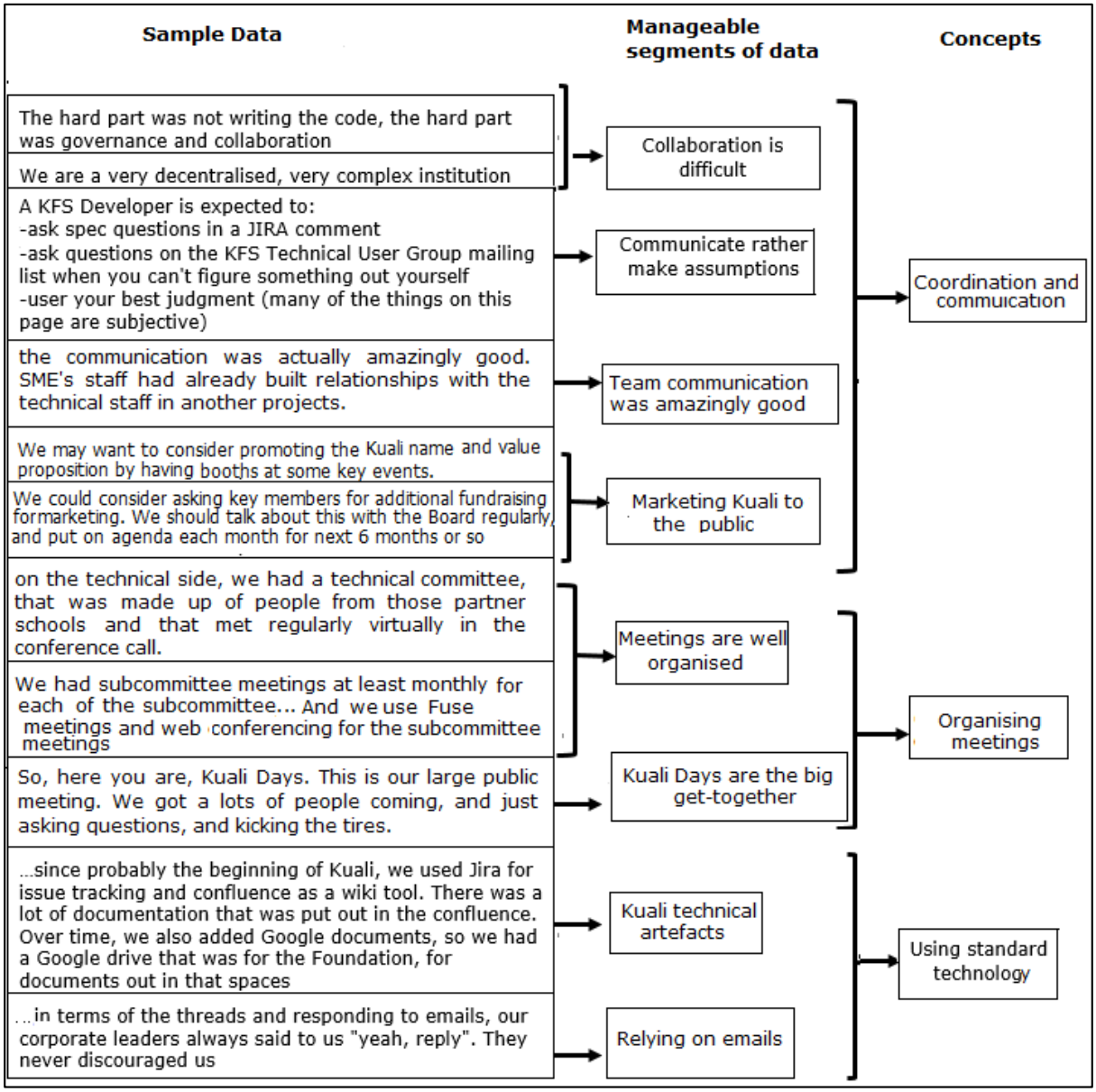
To sum up, the properties of the OSS code were emergent. Setting the properties of the code was initially driven by the experience with KFS project during the first governance phase. However, as the community evolved, these properties were altered due to the changes in the surrounding context. This illustrates that the OSS code is a resource that needs to be aggregated and distributed. It has properties that emerge through interactions.

5.3.4. Facilitating Collaborations

Balancing the interests was a boundary decision to embrace diversity and reduce the cost of governing the distributed resources. This was evident through the practice of facilitating collaborations, as shown in figure 14. The expansion of the community distributed the resources across multiple level of organisations: Kuali project, university, country, and Foundation levels. Coordinating distributed resources was one of the main governance challenges after the expansion:

> "*We sometimes pull in different directions. We have to be coordinated, and that's not so easy. It's like herding cats*" (S3)

Figure 14: Data Structure: Phase2-Setting Means of Collaboration



Therefore, the community members were urged to communicate rather than making assumptions. They were encouraged to attend face-to-face meetings (e.g. workshops, Kuali days) when possible, ask questions, and report any issues in Jira in order to get the community support. The following in an excerpt from one of Kuali documents:

> *"A KFS Developer is expected to:*
>
> *-communicate rather than make assumptions ("when in doubt, ask")*
>
> *-when assumptions must be made to avoid impeding progress, communicate those assumptions*

-raise any blockers immediately

-use the best communication medium available" (Document Name: Expectations of KFS Developer)

One of the major means of facilitating collaborations was through the use of standardised technical artefacts for coordination and communication. Tech3 summarises the main technical artefacts used during the second governance phase:

> "…*since probably the beginning of Kuali, we used* ***Jira*** *for issue tracking and confluence as a* ***Wiki tool****. There was a lot of documentation that was put out in the confluence. Over time, we also added* ***Google documents****, so we had a* ***Google drive*** *that was for the Foundation, for documents out in that spaces… We had* ***fisheye*** *for doing source code changes as well as doing code reviews*"

It was apparent that the technical artefacts were utilised differently among different projects. Initially, standard technical artefacts were imposed on all Kuali projects. However, as they were used by different stakeholders, their properties were changed. For example, the intention of introducing Jira was to act a bug tracking system that directs the work:

> "*...they [i.e. team members] will report bugs, but often there are times where we break them down to sub-tasks, the assignee is the person who is doing the work. And then a watcher is somebody who gets a notification when action of the Jira occurs...So, simply, everybody relevant is associated with the Jira*" (Tech2)

However, some user groups used Jira as an open forum:

> "…*but there have been time when Jira is just has been used as an open forum. If somebody found a bug and they said 'what is going on with this and how is it fixed, can you tell us about it'"* (Tech2)

Moreover, the teams utilised the tools according to the convenience of the user. The findings revealed that Jira and Wiki were used interchangeably in practice:

> "*Some of the subcommittee members and chairs are comfortable using Jira and actually put their enhancements in there. But, a lot of them are more comfortable using the Wiki*" (Tech5)

In addition, Kuali Wiki was a web-enabled tool that was used to archive and share project documentations. Project teams used it differently depending on the maturity of the project and preferences of the users. For example, figures 15 and 16 illustrate the interfaces of Kuali Wiki for KC and KS projects respectively. Since the KC project was built on a well-established community, KC project was well documented and thus the KC view in Wiki, as shown in figure 15, provides more extensive list of documents. On the other hand, KS project started from scratch, and thus less developed in comparison to KC project, which is reflected in KS Wiki home page (figure 16). Moreover, it is evident from KS home page that the users prefer more interactive interface.

Figure 15: Kuali Wiki - KC View

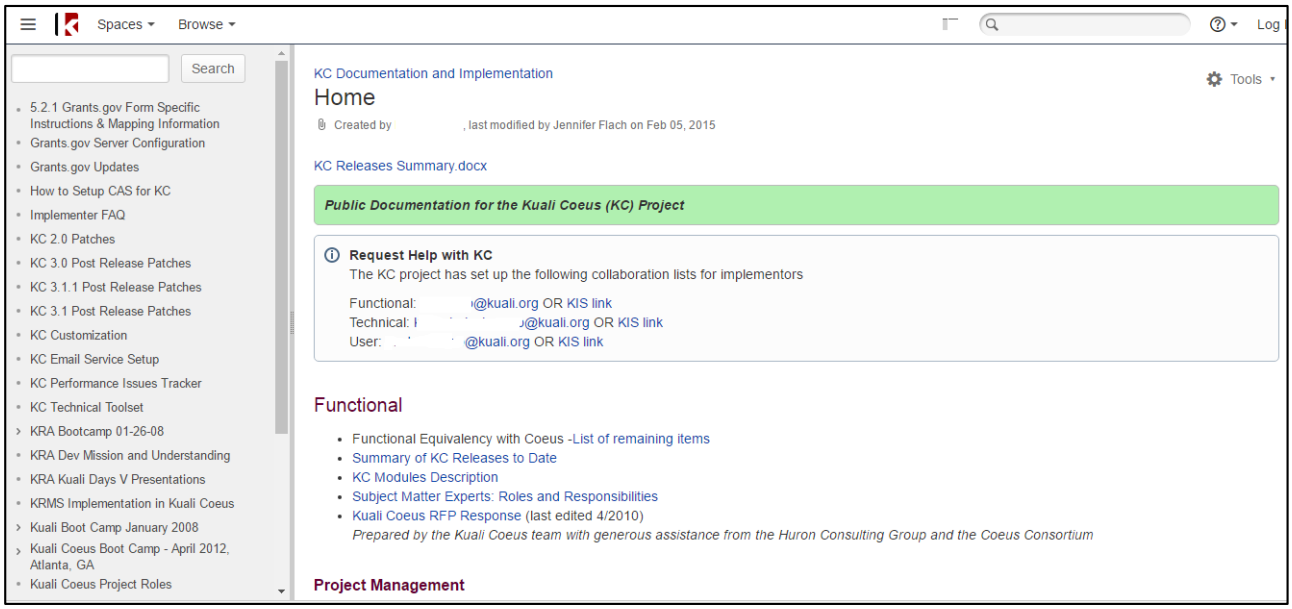
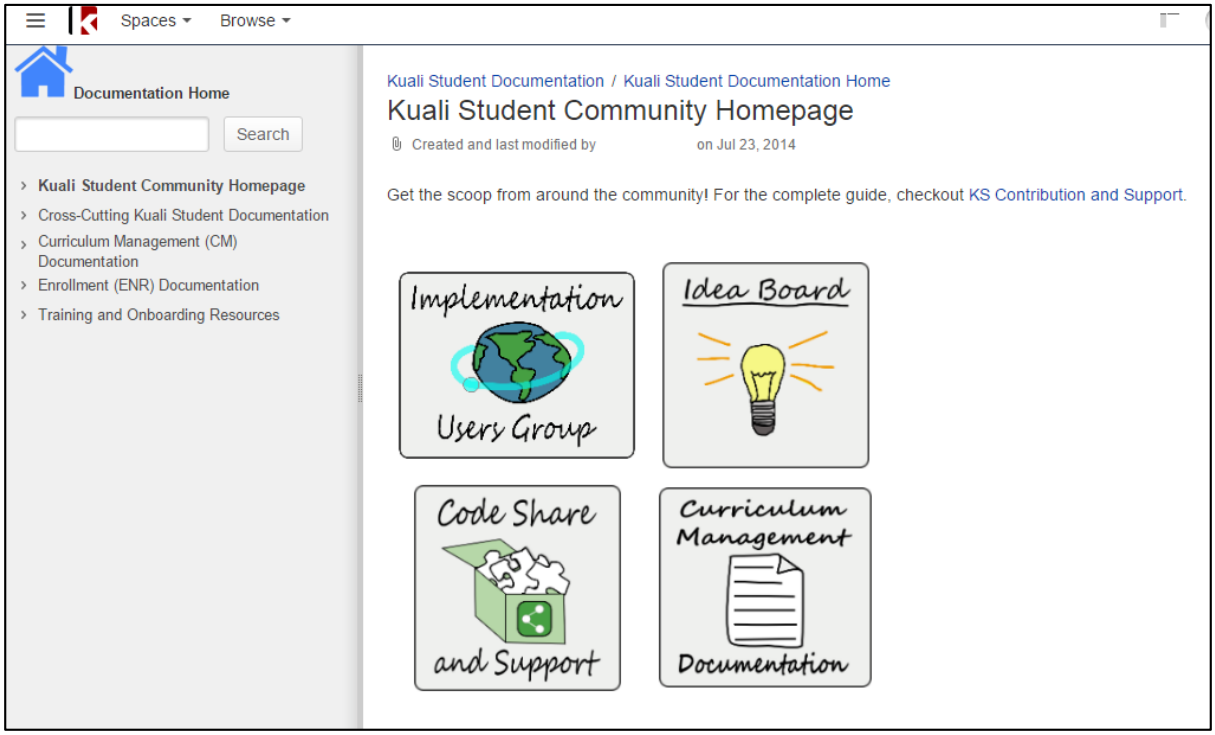


Figure 16: Kuali Wiki - KS View



Moreover, the use of the emails is also an example that illustrates that technical artefacts are not only mediators. The intention was to consider emails as the least reliable communication tool. This was evident in the documentations. For example:

> *"all communication mediums are not created equal*
>
> *we value Face to Face communication over Video-conferencing*
>
> *we value Video-conferencing over Voice-only communication*
>
> *we value Voice-only communication over Text chat/IM*
>
> *we value Text chat/IM over email*
>
> ***we value email over nothing***
>
> *Use the medium with the most appropriate bandwidth for the need"*
> (Document Name: Expectations of KFS Developer)

However, during the early days of the second governance phase, the community was heavily reliant on emails to ask for and provide support:

"*There were definitely times during the implementation when we reached out to the mailing lists and said 'hey we're experiencing this problem, have any of you has experienced that?' and we got good feedback and good help. We also used the mailing lists archive in a number of occasions to see if there is something out their when we ran into a particular problem*" (PM1)

Users across projects and countries heavily used emails to discuss generic and university-specific topics. In response to that high reliance on emails, the Foundation categorised the mailing lists into technical and functional emails to facilitate the discussions. Moreover, emails became a supportive tool for the other artefacts. The following are excerpts from email discussions where users refer to Wiki's and Jira's:

"*According to the ""Contributing Developer Responsibilities"" on the foundation wiki (https://wiki.kuali.org/display/KFSIMP/Contributing+Developer+Responsibilities), we are to include the JIRA number in the commit message. For contributions, do we need to wrap any of our changes in a comment with the same JIRA number?*"

"*Hi All, does anyone know the status of the above JIRA? We would like the ability to cut checks prior to their due date and was wondering if anyone is currently doing this?*"

The above illustrations suggest that the use of technology is not predetermined by its design. Instead, the properties of technology emerge in practice.

5.4. Consequences of Phase 2

During the second governance phase, Kuali expanded its scope to include diversity in order to enrich the community. Balancing the interests was a boundary decision that aimed to fulfil the interests of the diverse stakeholders on one hand, and control the community deliverables on the

other hand. The main governance practices were structuring the community, managing divergent interests, setting the properties of the code, and facilitating collaborations. These practices had two major consequences that triggered the need to reconfigure the community boundaries and spinout KualiCo.

First, the governance practices followed the community source approach. This has kept the software development process within the university boundaries, and thus the overall progress of the projects was slow and inefficient:

> "*...and that actually was our problem with Kuali, that our development, when we kept it with just higher education people, we were just too slow*" (Admin1)

Second, the restrictiveness of Rice framework and the different reactions from the Kuali projects were accommodated by expanding the base-code of Rice. Accordingly, the complexity of Rice has grown over time:

> "*When you have a system that is this large, it is more difficult to change things. So, it is more difficult for us to change certain tools and libraries that we have wanted to. So,* ***some of the complexity grows over time and we ended up with a lot of technical debts*** *overtime that becomes sort of expensive to change*" (tech3)

> "*…we've tried make Rice lighter easier more flexible, more easily adaptable in projects, but we never quite succeeded.* ***We made a whole sale break with the prior approach which led to go to KualiCo***" (Admin4)

As illustrated in the aforementioned quotations, the consequences of the tight control during the second phase led to spinning out of KualiCo. Kuali community had to rethink its strategy. This is further explained in chapter 6.

5.5. Conclusion

Balancing the interests was a boundary decision to aggregate and distribute community resources due to the emergence of multiple trajectories in Kuali community. This chapter showed that resources are not fixed entities with predefined properties. Therefore the governance practices that were evident during the second phase mainly focused on including the properties of the resources that are meaningful for a particular context, and excluding others. This illustration of governance practices deviates from the current views by acknowledging dynamicity. In this chapter, I also provided empirical evidence on the role of materiality in governing Kuali community. The properties of technology, OSS code and technical artefacts in particular, were rearranged to enable governance practices.

The research findings also revealed that controlling the code was the main desire behind forming Kuali community in the first place, and continued to be a desire to ensure the quality of Kuali projects. However, imposing Kuali Rice was the "*collateral damage*" (Tech6) that led to the demise of the OSS version of the product, as will be further explained in chapter 6.

Chapter 6: Kuali Governance Phase 3: Sustaining the Community

6.1. Introduction

During the first decade of Kuali's life, Kuali priorities were directed towards functional equivalence, with less emphasis given to the design of the system and user experiences. Besides, the development process was bounded by HE employees and HE rules. Accordingly, the overall progress of the projects was slow and inefficient. The community source approach also imposed the restrictive Rice framework that raised issues regarding the efficiency and sustainability of the community:

> "*…for the first several years, the basic question was 'can we get the software out the door in the first place?'… that means creating the essential core system... The questions after that… 2010/2011, have been much more on sustaining, building, and immigrating… we now instead of building and to manage and control a software of particular kind, we could try to find a way to achieve more speed of development and do it at lesser cost and using more common resources across all of the projects*" (Admin4)

Accordingly, the Board members decided to invite portfolio investors, which means dedicating certain investments to improve Kuali products; i.e. KFS, KC, KS, and Rice. The following is an excerpt from a Board meeting during December 2013:

> "*Portfolio investors might form a Board just like Project Boards* ***to articulate priorities about how to spend those funds****. It may need to be distinct because investors might not be members of this Board. But, those* ***investors shouldn't steer Kuali in way this Board would not endorse****… We could consider that Portfolio investors could have strong advisory rights… This Board should* ***have veto rights if plans aren't heading in the right direction****.* ***The***

> ***Portfolio investors could counter with the fact of the golden rule****. Maybe this Board would designate an overlap role, someone from the Portfolioness Board on the Foundation Board"* (Meeting38)

Inviting portfolio investors means enabling the community to get over the challenges of the first decade, in particular the golden rule and the restrictiveness of Rice. This new strategy aimed to direct Kuali projects towards improving the OSS products, instead of fulfilling what partner needs, as per the golden rule. The portfolio investors will also provide the community with separate funds to ensure a faster and a higher impact development process, in comparison to the slow steady progress during the previous decade. This suggests that the new direction of Kuali community became oriented towards improving OSS products. This required engaging with expertise outside the HE sector to assist in planning the new strategy:

> *"After much discussion,* ***we began moving toward the idea of hiring someone within the community*** *with leadership, communication, and organizational skills who can plan this, and consider titling it a 'portfolio-ness planner', and then* ***that person would hire a consultant as needed who has the professional expertise.*** *Since we are a non-profit, could we* ***leverage someone who has been at another non-profit so we have some outside perspective****"* (Meeting38)

This chapter explains the third boundary decision; sustaining the community, and how it has changed the boundaries of Kuali community. It has included and excluded actors, adjusted the roles, reconfigured the way technology is used, and the altered the communication patterns among the community members. This chapter represents Kuali governance practices from 2014 to 2016. It mainly focuses on how the authority to control the community has been transferred to KualiCo. During the data collection stage, Kuali community was in a transition period, and thus Kuali documentations, in particular rules and regulations, were not a reliant

source of information. The documentations mainly focused on the period before the inception of KualiCo. Therefore, the main sources of information for this empirical chapter were interviews, Kuali official videos, and the summaries of Kuali Board meetings.

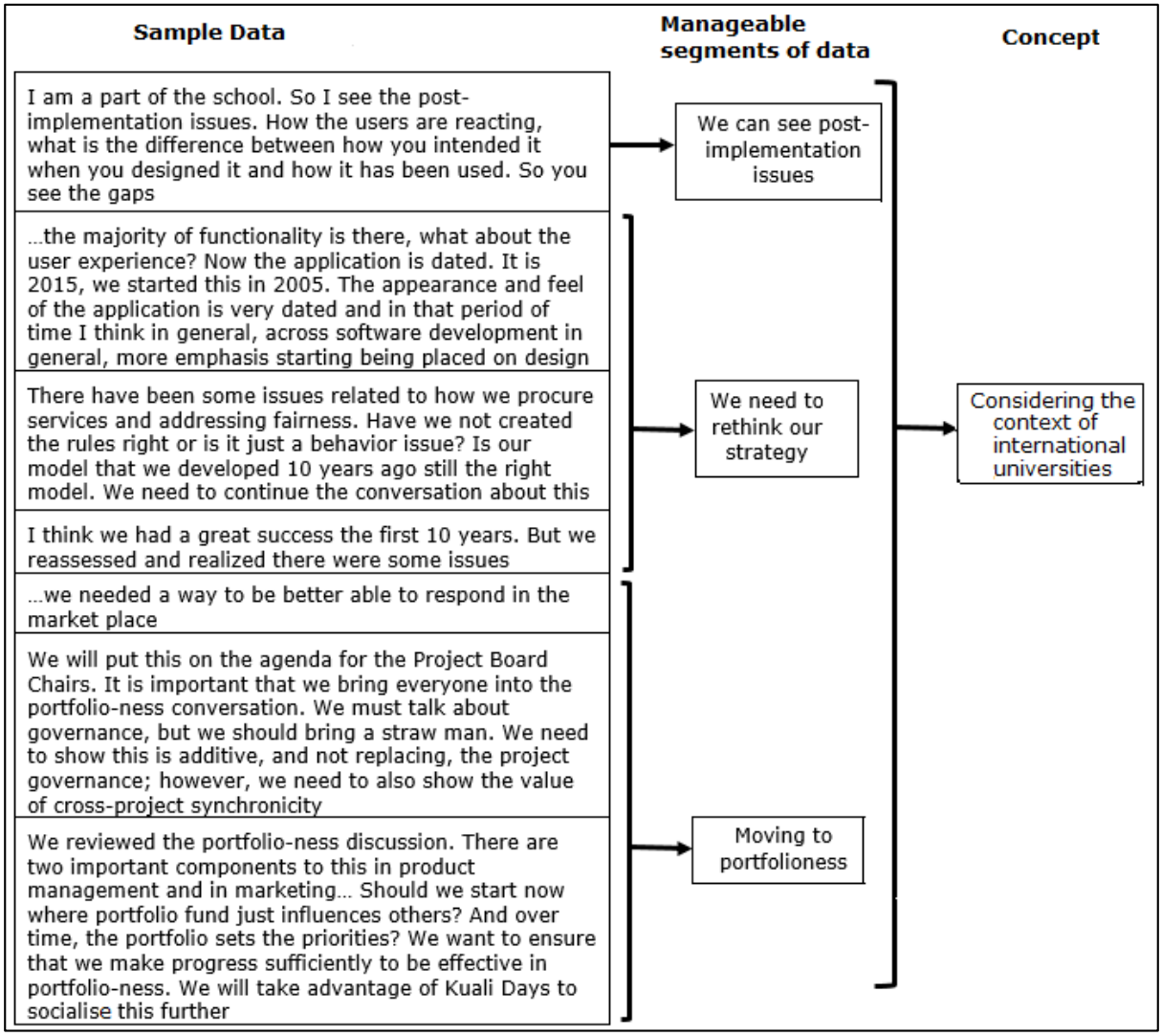
This chapter is structured in a similar way as chapters 4 and 5, which mirrors the axial coding paradigm (chapter 3). I begin this chapter, in section 6.2, by explaining the conditions that triggered the third boundary decision and moved Kuali community to the third governance phase. Then, I illustrate the governance practices of the third governance phase and their consequences in sections 6.3 and 6.4 respectively. Finally, the chapter is concluded in section 6.5.

6.2. Conditions of Phase 3

Prior the inception of Kuali, the processes of analysis and design focused on understanding the context of the HE sector in USA. After expanding the community, analysis and design also extended the focus to include the global requirements of the HE sector and the backgrounds of the ERP modules (i.e. KFS, KS, KC). However, during the second governance phase, the community experienced a series of stabilisation and destabilisation, and thus the community had to rethink its strategy. The following is an explanation of the conditions that triggered the third boundary decision; sustaining the community.

As explained in the previous chapters, the conditions emerged from the processes of analysis and design. Figure 17 visualises the process of moving from data to analysis and design with regards to the third governance phase, which mainly focused on the context of international universities. However, due to the profound changes that has occurred to the settings of Kuali community, I will explain these processes by first illustrating what has enforced changing Kuali strategy. Then I will explain the new strategy, which was known as moving to portfolioness.

Figure 17: Data Structure: Phase3-Analysis and Design



6.2.1. Rethinking Kuali Strategy

The second governance phase was about incorporating diversity to enrich Kuali projects. Thus, the Foundation has been engaged with multiple projects; however none of them was prepared for the next decade. It was evident that Kuali projects ended up in different stages of maturity (S2):

> "*There were big differences. The technology stack was different. The development standards were different. There were major differences between them… I think the Foundation never governed the fact that all the projects should be at the same technology*" (Tech4)

The Foundation was concerned whether they "might be stretching the Kuali community too far" and "being spread too thinly" (Meeting2):

> "***We need to rethink our strategy*** *on some of these key issues:*
> *--BI [i.e. Business Intelligence]*
> *--User Experience (not for the back-end users, but for the student/faculty facing users) --Technology*
> *-Cloud enabled*
> *--**Our current governance structure doesn't allow us to focus on overall strategic directions above and beyond the partner priorities**. How can we keep the "those who bring the gold makes the rules" value system and still focus on strategy?*
> *-Can we find resources to "go big?"*
> *-Can we go "university in a box" and "sell it"?*
> *We may need to do these projects, above-project, funded separately. Each project needs to get those artifacts that allow it to implement quickly and effectively. They would be separate projects, with separate expertise*" (Meeting24)

As illustrated in the above quotation, Kuali Board decided to rethink the strategy that was applied during the first decade of Kuali life. Since Kuali projects were developed by HE for HE, the Foundation was capable of viewing post-implementation issues to assess its strategy. This is an important matter because previous studies neglected the role of post-implementation in further developing OSS communities (Aksulu and Wade, 2010). As part of the analysis process, the Foundation Board members continuously compared between how they intended to build Kuali projects and how they were used, and thus identify the gaps. They have realised that the main focus was on the functionalities of the projects to ensure a vertical OSS community, where the functionalities were determined by Kuali partners, as per the golden rule.

Rethinking the current strategy revealed that the design of the interface, and the technology in use were given less priority. Besides, the community source approach and Rice framework hindered the progress of Kuali projects:

> "***The community is friendly but can seem impenetrable***. ***Some want a more traditional vendor relationship***. *Kuali requires CIOs and other leaders to figure out how to create a roadmap of what they want to do with their systems and be able to match their needs/skills/priorities with where Kuali and its projects are… --**We talk too much about technology or specific functionality**. **We should be focusing on the business benefit** to the school by joining a community that wants to improve the quality and cost of admin systems for higher ed*" (Meeting28)

The above quotation illustrates that Kuali had to introduce market-oriented concepts to improve Kuali products, rather than focusing on solving specific problems of universities. Therefore, Kuali Board members decided to invite portfolio investors. This is further explained in the next sub-section.

6.2.2. Moving to Portfolioness

The aim of inviting portfolio investors was to improve the OSS products. Therefore, the processes of analysis and design at this stage were focused on gathering the requirements that will improve the products. This was done by analysing the status of Kuali projects and what are the main requirements to move on to the next decade.

As mentioned in chapter 5, Kuali projects followed the golden rule, and thus the partner universities set the requirements of each project. However, the new Kuali strategy included the needs of the software market to sustain Kuali community:

> "*And we were not able to generate the commercial penetration that was necessary for the survival of the software*" (Admin2)

In addition, during the first decade, Kuali community adopted a community source model. This means that Kuali contributors were identified, task distributions were clear, and project members agreed on work flow processes. The development process in Kuali community was similar to the corporate world in terms of the organisation. However, improving Kuali products required more competent human resources and efficient development tools, which did not exist in Kuali community:

> "*I would say that it [i.e. Kuali] is similar to corporate method of software development, with the only difference being that the culture in higher ed in the united states is much more accepting of slower time frames, less human resource that might not be a calibre that you would find in a corporate world*" (Admin1)

Moreover, following a community source approach, the partner universities were in control of the projects by setting the priorities of each project. Inviting portfolio investors to improve the products raised issues with regards to the domain of control and influence, and the role of the community in particular:

> "*Should we start now where portfolio fund just influences others? And over time, the portfolio sets the priorities? We want to ensure that we make progress sufficiently to be effective in portfolio-ness*" (Meeting39)

This has called for a shift in the community boundaries to reconfigure the governance model that was based on balancing between growth and control. This has triggered the third boundary decision, sustaining the community, and accordingly the community entered the third governance phase.

6.3. Governing Practices during Phase 3

Sustaining the community was a boundary decision that focused on determining the domain of control and influence in Kuali community.

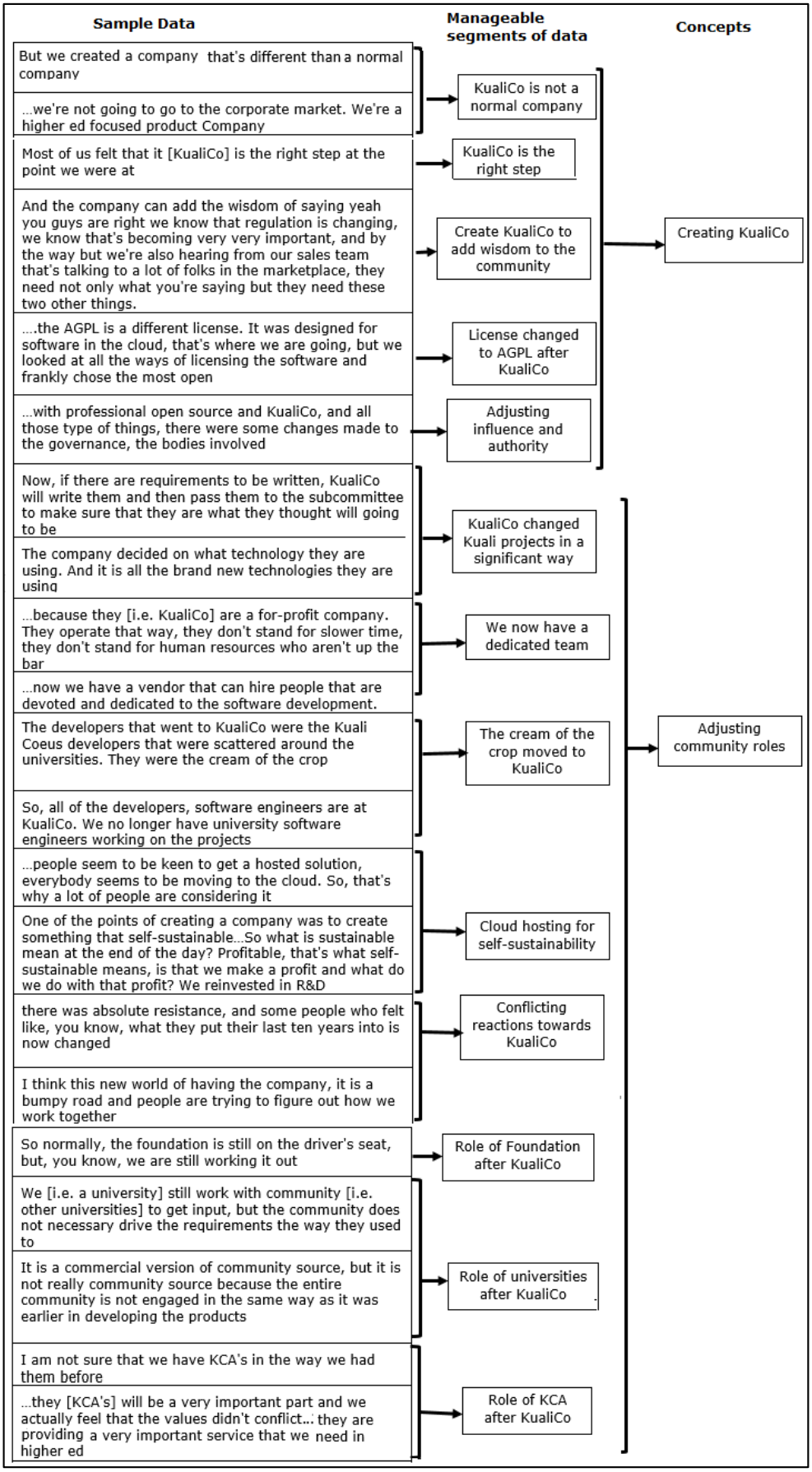
Previously, the partner universities were the influencers and controllers of Kuali projects. However, introducing market-driven concepts raised uncertainty with regards to the future direction of Kuali projects. Therefore, Kuali Board members decide to spinout KualiCo, a for-profit company, to be responsible for the code development process. However, this has questioned the role of the Foundation, the partner universities, and KCA's.

In this section, I illustrate through empirical evidence that the dynamicity of Kuali community produced intended and unintended consequences that contributed to the change of the settings of Kuali and the surrounding context. Therefore, the community continuously changes its boundaries, and thus its governance practices, to cope with these changes and govern the collaborative efforts. I also illustrate how KualiCo rearranged the material properties of the technology to meet the objectives of the third phase; improve the products. There were two salient governance practices during the third governance phase; restructuring the community, and reconfiguring coordination and communication.

6.3.1. Restructuring the Community

Sustaining the community was boundary decision to improve Kuali products and prepare them for the next decade. This objective created profound changes in the context of Kuali as it moved Kuali beyond the desire of fulfilling what universities need. Therefore, sustain the community entailed restructuring Kuali community, which involved creating KualiCo and adjusting the community roles, as shown in figure 18.

Figure 18: Data Structure: Phase3-Restructuring the Community



6.3.1.1. Creating KualiCo

KualiCo was formed in August 2014 to control the code development and to restructure the Kuali community:

> "*So, in 2014 we re-imagined what this community and what Kuali could be like. And what we decide to do was to create a company that combines the best parts of the community with the best part of a company*" (S8)

KualiCo is a higher-education-focused company that is providing OSS solutions and selling implementations as cloud services. It was formed as an improvement to the idea of inviting portfolio investors. KualiCo was formed to control the software development process; however it was not a conventional company:

> "***We [i.e. KualiCo] are not taking private equity, we're not taking venture capital, we're not taking investment outside***. *And that means we're not beholden to Wall Street, we're not beholden to quarterly profit calls, we're not beholden to cost cutting just because we want to squeeze out that little last dollar for investors. We don't have to worry about that. The owners of the company are the Foundation and the employees*" (S8)

The idea of KualiCo was, rather than inviting investors to develop Kuali products, universities aggregate their resources

> *"then they can take those aggregated resources and in a single contract with Kuali the company, which arranges for how they will be spent? What the priorities will be? What the governance of it will be? Then you have a single entity in the foundation and a single entity in Kuali the company that can execute contract with each other rather than having 13 or 15 different contracts in all of our procurement officers and lawyers helping with that"* (S10)

The founders of Kuali community and the individuals who witnessed the development of Kuali believed that KualiCo was the right step. Creating KualiCo was not due to loss of belief in OSS route (Admin2). Instead, the aim of forming KualiCo was to focus on improving the products that were experiencing growing complexities:

> "*I can't speak for all of the KFS community, but as someone who has been around from the beginning and has witnessed all of our growing pains, I'm confident that this is the right next step and we will continue to be functionally driven, even if our software is made available in different technical ways*" (Email, Nov 2014)

In addition, KualiCo was created to add wisdom to the community. Kuali community comprised of research-intensive universities as well as small colleges. The community needed "*the voice at the middle*", as explained by S10:

> "*If we look back and reflect on our prior decade, the folks who brought the gold if you've heard us say golden rule before, we brought that money together… we built a software that we needed. It fit what investors needed to be done. But there was no one sitting at that table that was advising what does the rest of higher ed need*"

KualiCo became the controller of the community. It provided services on-premises and on the cloud. Accordingly, the license of the Kuali products have been changed to AGPL license, which is a restrictive OSS license designed for OSS in the cloud (Admin2). Cloud hosting services are offered with a license fee. The AGPL license restricts derivative work to be published under the same license.

The inception of KualiCo changed the governance structure of the community, and accordingly altered the sphere of influence and authority. The research findings revealed that KualiCo took over the control of the future direction of Kuali projects by introducing

> "*a new development discipline and a more streamlined decision making process that resulted in a more robust software*" (Admin6).

This was achieved by changing the structures of the projects. For example, PM1 explains the changes that has occurred to KFS structure:

> "*So previously, there had been what was called the KFS Board that was operated at the higher level, and then the KFS functional council that operated below the board, and then even below the functional council were a bunch of Subject Matter Experts subcommittees... And as part of this transition, we really looked at how we could streamline that to improve our communication, improve our agility, and basically collapsed the functional council and the functional subcommittees into a group called the Customer Advisory Group"*

Forming KualiCo and granting it the authority to control has also eliminated the voting system that was used by project teams:

> "*One thing that has changed is that we don't have that formal voting structure within the subcommittees now. So, the subcommittees are still providing a lot of input into our enhancements and providing guidance on how we roll out different things. But, they are not the only voice*" (Tech5)

Accordingly, project teams lost their authority to decide on the future direction of their projects. Admin7 explains the situation in KC project:

> "*The KC community is very used to having its say in what happens next and we are not used to having another group that is also informing what is happening to the software*"

This suggests that the formation of KualiCo restructured the community. This had salient implications on the roles of the community members. This is explained in the following sub-section.

6.3.1.2. Adjusting Community Roles

The main objective for creating KualiCo was to improve the OSS products regardless of the specific needs of the universities. KualiCo has a dedicated team to meet the new objectives. This has raised enquires regarding the roles of KualiCo, the Foundation, and the universities. Besides, having KualiCo as a controller questions the role of KCA's in the community. The following are illustrations on how KualiCo introduced new roles, and adjusted the roles of the community members.

Role of KualiCo

As mentioned previously, KualiCo became the controller of the code development process. It has changed Kuali projects in a significant way. KualiCo transformed Kuali projects from complex dependent ERP modules to decoupled individual services, where the latter refers to controllable pieces of code:

> "*We're* ***moving away from having just a monolithic suite****… to instead a set of* ***individual services.*** *We've made great strides in* ***refactoring our financial product and our research product***" (S8)

The highlighted parts of the above quotation emphasise on the materiality of the OSS code. During the previous phase, restricting the contributions was the main desire, and thus the code was designed in a self-regulating manner; developers had no choice other than following the code. On the other hand, the third governance phase aimed to improve the efficiency of Kuali products, and thus the code was redesigned accordingly.

Moreover, while KS was rewritten from scratch (again), KFS and KC were refactored to decouple their modules. Refactoring means changing the internal codes of the software without changing the overall functionalities. This was achieved by negotiating the changes with the corresponding project teams in the universities. In addition, KualiCo has replaced Rice

with CORE, where the latter is a less restrictive framework to build loosely coupled services:

> "*Kuali Core is a set of independent services that would… support all of the different products... So if you just want to use the off-service or if you just want to use the workflow service, you'll be able to use those independent services*" (S9)

Another major role of KualiCo was improving the users' experiences. KualiCo dedicated designers to improve the look and feel of Kuali projects, especially KFS, which had an outdated interface:

> "…*one of the really good things that when the KualiCo company was formed, they* ***hired a designer for the Financials team****, and that designer has been working with users to try to improve the users' experience. Starting with the main dashboards sort of portal, and then move on to purchasing and travel as well.* ***But, it is definitely different user communities with different needs***" (Tech3)

Besides, KualiCo focused on smaller, but specialised, teams of high performing individuals (Tech3). These dedicated teams of KualiCo initially consisted of former developers that were distributed among partner universities. "*They were the cream of the crop*" (Admin3). Accordingly, universities no longer have developers working on the projects (Admin7):

> "*KualiCo needed to steal, I hate to use that word, to steal those people from the universities because… they knew KRAD because we still have to maintain that code until we get rid of it; they know KNS… but also because they were the smart smart smart people that will be able to pick up all the new technologies*" (Admin3)

The findings revealed that KualiCo played the functional and technical roles. It communicates with universities as one-to-one to understand their requirements. Then, KualiCo decides on the priorities and the type of technology that will move the product to the next level (Admin7, Tech4).

Besides its main responsibilities that are directly related to the code development, KualiCo is also responsible for handling the associated technical and administrative burden.

Moreover, KualiCo offers cloud hosting services along with the on-premises services to ensure self-sustainability, which was one of the top priorities of Kuali during the third governance phase. The cloud hosting capability emerged from the community's desire to have a self-sustained company, and thus less reliant on external entities. This is important to protect the specificity of the HE sector. Cloud hosting was designed and reconfigured according to the needs of the universities:

> "*when we sign a new cloud contract what do we do with the money that we make? We don't pay a venture capitalist or private equity person,* ***we go hire more developers and designers****. Every time we sign a new cloud customer…we take that money* ***we reinvested in research and development*** *which helps every one of you*" (S8)

Besides, the cloud hosting was an attractive solution, especially for the universities that lack the infrastructure and support to maintain a complex ERP system locally:

> "*having it in the cloud, having somebody else manage it, having somebody else dealing with the upgrades and all of that, in the context of a more responsive relationship than what you would have with a profit-driven vendor, we think is the way of the future and the way we should go*" (Admin6)

This suggests that KualiCo with its dedicated services, including cloud hosting, will advance the projects and create a thriving OSS community:

> *"…at the point of the move to the KualiCo partnership model, we felt that we were optimistic to be able to fix some of the constant delays, the inability to meet deadlines, the inability to produce functioning*

code. We felt ***it [i.e. KualiCo] was a result of an unclear capacity to make decisions to actually move forward***" (Admin6)

"***KFS has stalled in terms of adoption, probably for a couple of reasons that we, as institutions are not equipped to address*** – *such as marketing, providing a complete suite of software, leveraging what makes us unique, our community, and offering a competitive cloud option to compete with other software providers*" (Email, Nov 2014)

Despite the evident advancements in the progress of the projects, the formation of KualiCo caused unintended consequences. There were conflicting reactions towards authorising a profitable company the right to control Kuali projects:

"*So with the event of a KualiCo… we were I would say confused. There was a lot we didn't understand, there was a feeling of, you know, what are we losing? What are we gaining? So that was very unclear*" (S11)

"*…the inclusion of the company kind of goes back around to the first step… We were kind of in the path saying 'let's do it ourselves, let's not have vendor engagement', and now we have that. I think that has been mixed for some.* ***We did lose some partners because of it****. Because they felt that the ideals of open source were not being met*" (Admin7)

In the quotation above, Admin7 illustrates that loosing partners was one of the unintended consequences of forming KualiCo. One reason for this might be that the discourses about the new Kuali strategy, i.e. moving to portfolioness, focused on pure technical and business considerations. These discourses neglected the functional requirements of the partner universities, which suggested that their effort in building a community source is at stake.

Role of Kuali Foundation, Universities, and KCA's

The formation of KualiCo required adjusting the roles of the Foundation, partner universities, and KCA's. The Foundation had a central role. It had a permanent representative in KualiCo's Board of Directors. The representative was granted an access to KualiCo's details, including financials:

> "*The preferred director of the Kuali foundation can prevent us [i.e. KualiCo] from being sold. It can prevent us from going public. It can prevent us from trying to change our open source license. You need to know,* ***when we created this company these are intentions anyway***" (S8)

With regards to the universities, their major role was to propose their requirements to KualiCo, where the latter is responsible for validating and prioritising them. Since Kuali projects continued to be open source, universities were allowed to modify the code internally. However, they were not eligible to contribute the code back to the base. This means that universities lost their capability to control the code:

> "*We [i.e. universities] talk about ourselves as partners, but really… we don't have control, we just have greater influence. Whereas previously we had control, we were definitely in control under the old model*" (Admin6)

Regarding the role of KCA's, the community was not yet clear about the future of the KCA's at the time of data collection. However, the Foundation assumed that KualiCo will not be able to replace some of the existing roles of the KCA's:

> "*So we still need to have partners. We may change them from Kuali commercial affiliates to another name, we didn't decide yet. But we still need that, because whether someone decide to do a local implementation of one of our systems, just as they do now, it still*

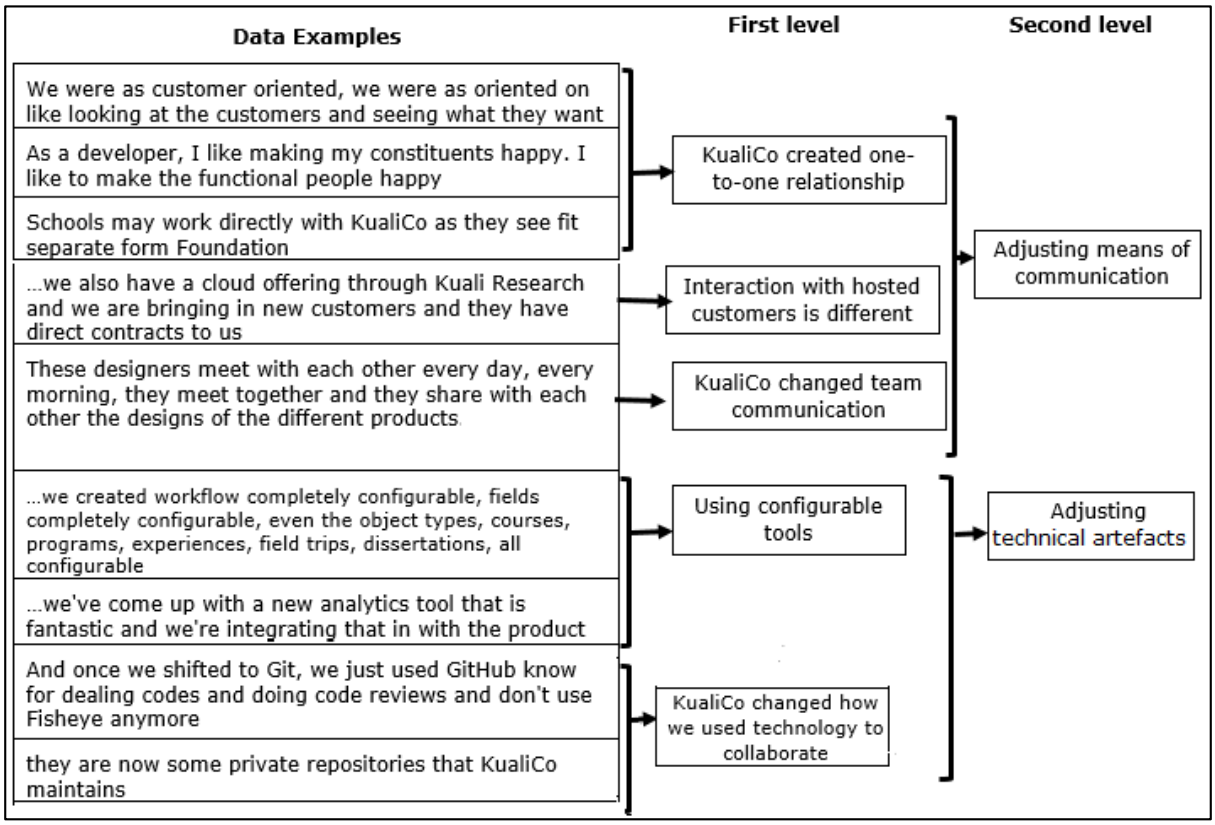
possible, even if KualiCo is there. Or whether they choose to buy the cloud solution from KualiCo. All of those needs still exist. Training, conversation, integration with your current systems, KualiCo is not going to do that. So, we still need commercial partners in the ecosystem institutions can work with to help them do that" (Admin1)

To sum up, in the new settings, KualiCo became the controller, the universities are the influencers, the Foundation is the conveyer, and the KCA's are the supporters. With this new settings, the community had to reconfigure the way it coordinates its collaborative effort.

6.3.2. Reconfiguring Coordination and Communication

Reconfiguring coordination and communication was a salient governance practice during the third governance phase. As summarised in figure 19, it was achieved by adjusting the communication patterns between the community members. It was also achieved by either rearranging the properties of the technical artefact used for communication and code development, or by introducing new artefacts.

Figure 19: Data Structure: Phase3-Reconfiguring Coordination and Communication



Adjusting the means of communication involved replacing the functional councils with a one-to-one relationship. The former was a collective approach to gather the requirements of Kuali projects prior to the inception of KualiCo. The latter is similar to a customer-vendor relationship:

> "*it is a one-to-one sort of relationship as oppose to the functional council model which was much more integrated except it was so* ***process heavy that we often refer to it as the dysfunctional council***" (Admin6)

> "*What we have done since the forming of KualiCo is our business analysts sit down with customers and ask them questions, go through their processes, and figure out what they are doing. So now, we can make it easier for them, we can make them better at their jobs, we can improve the product. We didn't have that kind of orientation before becoming the company*" (Tech2)

The above two quotations illustrate that the functional approach was process-heavy; hindering the development of the projects. Thus, it was replaced by a more efficient approach. However, it is evident that in the one-to-one relationship the collaboration between university partners has vanished. This suggests that the community source has also vanished. Universities have the right to communicate directly with KualiCo, separate from the Foundation, as required (Meeting49):

> "*Institutions then may, if they wish, choose to contract directly with Kuali the company if they want cloud services. So they may download the code and decide they want to run it on their own or they may say 'you know we'd like to just pay KualiCo to run this'*" (S10)

The research findings revealed that the collaborations between KualiCo and the customers, i.e. universities, varies depending on the service:

> "*…one hosted customer may do things very similar to one Kuali community member. So, it is not like they are split in any particular way.* ***It is just that the way we interact is slightly different*** *in that hosted customers we are supporting them directly versus the on premise customers, we give them enhancements, we give them a new release every month. But, they are still customising things locally and it is a different structure*" (Tech5)

As mentioned previously, improving user experience was one of the main roles of KualiCo. Therefore, KualiCo created the Experience Centre to facilitate the interaction across Kuali projects. In the below quotation, S8 briefly describes the communication between the designers:

> "*One of our first two employees was an interaction designer, and we now have four designers. One designer per product…* ***These designers meet with each other every day, every morning, they meet together and they share with each other the designs of the different products****. So financial share with student,*

student shares with research, and they comment on each other's designs"

The emphasised part of the above quotation illustrates that the Experience Centre has introduced more frequent face-to-face meetings, which was not evident during the previous two phases. In addition, this quotation also highlights a new communication pattern among team members. Previously, the communications were between the members of the same project. However, the Experience Centre required teams across projects to communicate.

The formation of KualiCo also adjusted the properties of the technical artefacts. This was evident in KS project because it has been rewritten from scratch, and thus KualiCo decided on the technology and the approach to be used in implementing KS project (Tech4). In addition, adjusting the technical artefacts was evident in the process of internationalisation. The global context of HE and the technology in-use reconfigured KS development approach. Instead of negotiating the best practices among the universities, as the case in the previous phase, KualiCo encouraged universities to accept their uniqueness and reject any attempt to change their internal work practices to cope with the KS project:

"*The International pieces are really important piece for us as well. So Student [i.e. KS] from the very beginning was written with international in mind*" (S9)

"***Why should we, a software vendor [i.e. KualiCo], force you to drive internal change for something that doesn't matter for the sake of saying that we're the same****. Now I'm not saying that we should not have best practices we should absolutely have best practices****, but it's more important for us to create a configurable tool that allows people to create things the way they want****… so that's why we created workflow completely configurable, fields completely configurable, even the object types,*

courses, programs, experiences, field trips, dissertations, all configurable" (S9)

As illustrated in the above quotation, KualiCo introduced configurable tools. The idea of configurable tools differs from the configurable code that was evident during the second governance phase (see section 5.3.2). The latter refers to pieces of code that are parametrised, such that they can be turned on or off depending on the requirements of the universities. On the other hand, the configurable tools are more like gadgets that allow universities to set their preferences without having to change the software code. These tools enable the universities to set their requirements, and KualiCo control the software development process. S9 briefly explains the use of configurable tools in KS project:

"*The forms are completely flexible. So you have the ability to change the forms anyway you want to.* ***So all of the fields, 100% of the fields on the forms are configurable, literally a 100%.*** *If your courses are crazy and you have some weird way of doing it, let's say you don't have the notion of credit hours at all in your university, you can make the forms reflect that. If you don't use learning outcomes or if you do use learning outcomes, that's completely configurable*"

In addition, KualiCo also altered how communication tools were used. Tech2 summarises these changes:

"*Since October 2014, we have moved to Git [i.e. GitHub][16]. We have moved away from Fisheye … We still use Jira, and we still use it in a very similar kind of way. We have moved to an internal cube. … So, basically it is a little bit less transparent because we are more concerned about competitor as we used to be in the past"*

Regarding meetings, Tech2 continues:

[16] A web-based repository of open source projects.

> "*the other shift that I should mention is in KualiCo we use SLACK… Which is basically a kind of a Skype, except it is a web-based and it keeps all the messages for ever and it allows different people to communicate. So you can basically create a channel that different people join and they are always on that channel together and go back and search messages and stuff*"

The above two quotations by Tech2 highlight how the properties of the technical artefacts have been reconfigured to cope with the change in the context. The internal cubes of Jira and the special channels of SLACK were reconfigured to add security and privacy. These "offline" conversations were evident in the relevant literature (e.g. O'Mahony and Bechky, 2008). However they referred to firms intentions to exploit the community's efforts, which violated the OSS ethos. In contrary, private conversations between KualiCo teams enables KualiCo to control the code and sustain the community, which is the objective of the third governance phase.

6.4. Consequences of Phase 3

The significant adjustments that had occurred to the community structure and the means of collaboration were due to the tight control that was imposed during the second governance phase. However, the restructuring of Kuali community during the third governance phase transformed Kuali community from community source to a customer-vendor model. The intention of this shift was to treat every university as special. This might sound as the right direction conceptually; however "*how do you get everybody who thinks that they are special to look for the common good*" (Admin6). This means that the third governance phase has transformed Kuali from a pure community source to a market-driven OSS community.

Another major consequence was with regards to the processes of analysis and design. Throughout the previous phases, the processes of analysis and design were performed collectively by connecting with the community members to understand what universities need. However, during the third

governance phase the focus was redirected towards improving the products. Thus, it was necessary to understand the needs of the market. Accordingly, the processes of analysis and design have been assigned exclusively to KualiCo, which is considered as a major change in the settings of Kuali community.

6.5. Conclusion

In this chapter, I illustrated that Kuali community is in a continual process of becoming. The structure of the community is fluid. It emerges through collaboration. Therefore, the collective effort is governed by continuously changing the boundaries of the community to cope with the changes within the community on one hand, and the changes in the surrounding context on the other hand.

Sustaining the community was a boundary decision to cope with the introduction of market-oriented concepts to Kuali strategy. This boundary decision altered the boundaries of Kuali by including KualiCo as the controller of the community. Accordingly, the salient governance practices during the third governance phase were focused on restructuring the community and the communication patterns among the members.

Most importantly, the chapter highlighted control as the main governance practice throughout Kuali life. The data revealed that controlling the code was the desire behind creating Kuali in the first place. Control was also behind imposing the restrictive Rice framework and its growing complexity. Moreover, control was also the reason to spinout KualiCo, and thus transformed Kuali community into a different community.

Chapter 7: Theory Development and Discussion

7.1. Introduction

Governing OSS communities ensures that the OSS community remains viable despite its fluidity and dynamicity. Therefore, governance means possessing and distributing resources on one hand, and adapting to the changes that occur to the overall context on the other hand. The story of Kuali provides insights on how the community governed the collective efforts through changing its boundaries overtime; i.e. boundary decisions. Drawing on the empirical findings, I have developed a model to conceptualise OSS governance through boundary decisions.

In this chapter, I describe the process of developing the emergent grounded theory from the empirical data that are represented in chapters 4, 5, and 6. This is done by elaborating the OSS governance model that has been developed during the selective coding phase and introduced in section3.5.3. In this chapter, I also locate the emergent theory within the current literature. This is an essential theory building step to extend and refine existing knowledge (Corbin and Strauss, 2008) with regards to OSS governance.

The data analysis and findings illustrate that the governance practices emerge from the interactions between four main constructs: context, control, resources, and materiality. I explain each of these constructs in section 7.2. Then, I explain the emergent governance model in relation to the empirical data and the current literature in sections 7.3 and 7.4 respectively. Finally, I evaluate the research process in section 7.5.

7.2. The Emergent OSS Governance Model

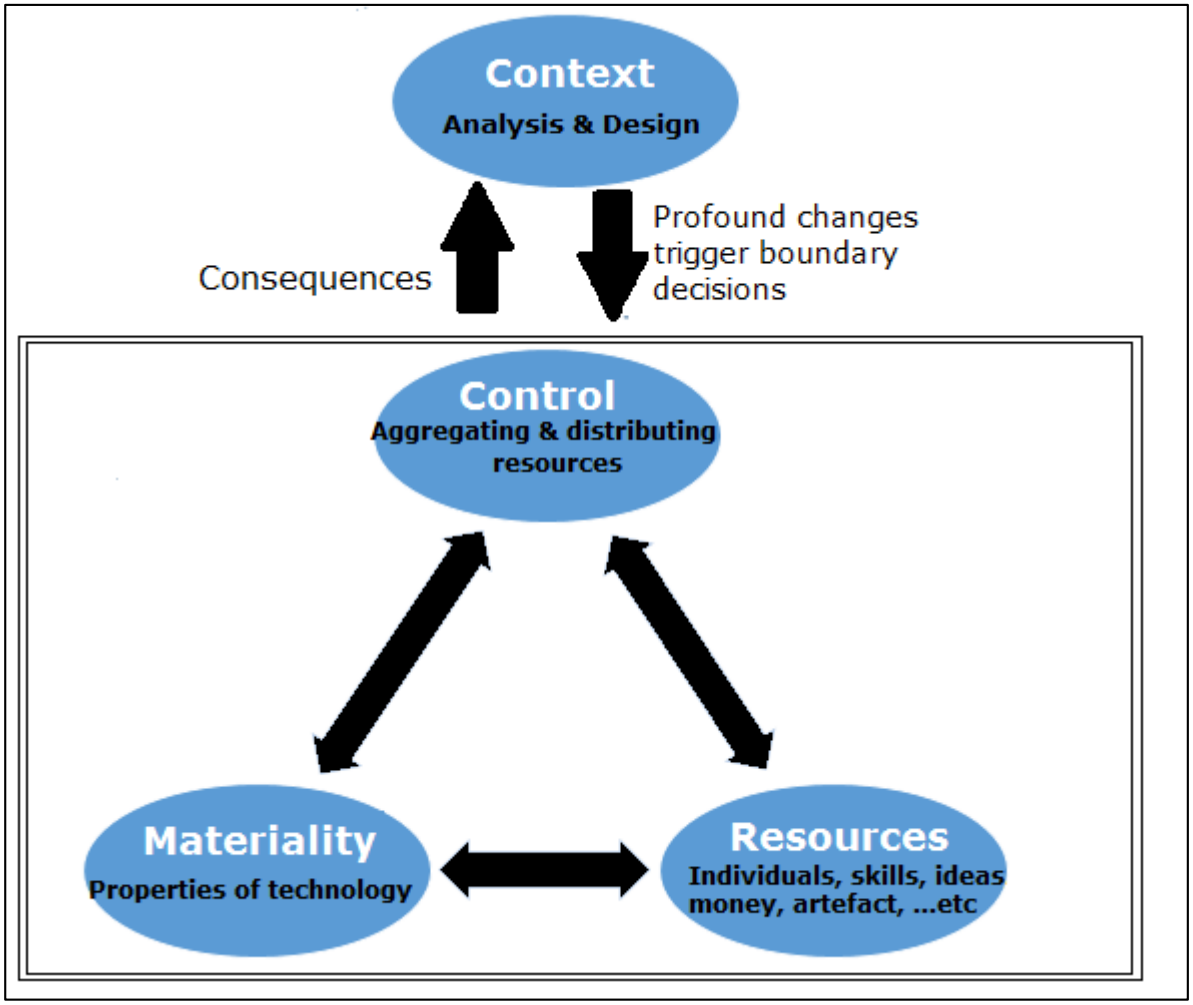
The main objective of this thesis is to develop a substantive theory to explain OSS governance. A theory, in general, is an interrelated set of well-developed concepts that can be used to explain a particular phenomenon (Corbin and Strauss, 1990). A substantive theory is a "*theoretical*

interpretation or explanation of a delimited problem in a particular area" (Charmaz, 2014:344). Although substantive theory is grounded in research in one particular substantive area, it may have implications and relevance to a wider context and become a stepping stone to the development of a formal grounded theory (Glaser and Strauss, 1967). In this section, I briefly explain the main constructs of the proposed OSS governance model.

The research findings represent OSS governance in a way that departs from the current views by encountering two main, yet neglected, aspects of OSS governance: dynamicity and materiality. In this thesis, I argue that OSS communities are governed through boundary decisions that identify the actors, resources, and domain of action to accomplish a particular governance practice. The decisions change the boundaries of the community to control the community and adapt to its dynamicity.

The role of boundary decisions has been discussed in the relevant literature in terms of promoting innovations in online communities (e.g. Jarvenpaa and Lang, 2011; Lauritzen et al., 2013; Teigland et al., 2014), how materiality of technology enables and constrains boundary decisions (e.g. Akoumianakis, 2014), and the role of boundary decisions in facilitating knowledge collaboration in online communities (Faraj et al., 2011). However, theorising boundary decisions as a governance mechanism in OSS communities and their consequences is an underdeveloped area of research (Akoumianakis, 2014).

Figure 20: The OSS Governance Model



Based on that, I have developed a theoretical foundation that acknowledges that OSS communities are in a constant state of becoming. I demonstrate that the OSS governance practices are emergent, contextual, and temporal. In order to explain the emergent theory, I have developed an OSS governance model, as shown in figure 20. The OSS governance model represents boundary decisions. A boundary decision is preceded with analysis and design to contextualise governance practices. This is represented by the construct of context. The main construct of a boundary decision is control, which consists of governance practices that aggregate and distribute resources. These practices are accomplished by the interaction with the available resources and the materiality of technology. In the rest of this section, I explain the theoretical constructs of the emergent OSS governance model.

7.2.1. Context

When explaining governance practices in OSS communities, scholar tend to situate the community within a wider context (Aksulu and Wade, 2010; Lauritzen et al., 2013), where the governance practices are bounded by an overarching social context (Winter et al., 2014). The current literature illustrates that the surrounding context is not passive. Instead, each governance practice is associated with non-governance-related technical and social practices that are context-related. Besides, the output of these practices are also inputs to others (Crowston et al., 2012).

The literature on OSS governance shows a growing interest in understanding how governance practices emerge in OSS communities (e.g. Demil and Lecocq, 2006; O'Mahony and Ferraro, 2007; Germonprez et al., 2014). However, current accounts do not provide sufficient explanation on which governance practices work together and in what context (Di Tullio and Staples, 2013). The research findings contribute to the current literature by emphasising on contextualising the governance practices. Here, I first explain what is meant by contextualising the practices and how contextualisation is essential to maintain the community identity and legitimise the governance practices. Then, I explain that contextualisation is performed through the processes of analysis and design.

Contextualising governance practices means adapting them to the growing complexity of the OSS community (Aaltonen and Lanzara, 2015). Based on an earlier study conducted by Fitzgerald (2006), OSS communities have been transformed to a new generation that introduced OSS to domain-specific industries, which generated goal-specific contexts. Therefore, scholars (e.g. Crowston et al., 2012) called for attending to the context in order to identify the governance practices that are more amenable than others in that particular context.

Contextualising governance practices is important to maintain the identity of the community by defining the objectives of the community (i.e. who

they are) and achieving coherence between the mission of the community and its activities. OSS communities are described as identity-based communities because the members collaborate towards a common goal and feel committed to it; i.e. members are attached to the community as a whole (Ren et al., 2007). While conventional organisations maintain sense of identity to create competitive advantage over others, maintaining the identity of the OSS community is essential even in the absence of direct competitor (Chen and O'Mahony, 2009). Community members that share a sense of shared identity have more willingness to collaborate and more likely to achieve the common objectives (Jarvenpaa and Lang, 2011).

However, the identity of OSS communities can be ambiguous because resources can easily flow in and out of the community (Ren et al., 2007; Watson-Manheim et al., 2012). This means, it is challenging to maintain a sense of community among unknown diverse community members (Markus, 2007). In the current literature, scholars argue that the identity is maintained through governance practices, such as social practices (Markus et al., 2000), OSS license (Ojha and Rao, 2014), and memberships (West and O'Mahony, 2005; O'Mahony and Bechky, 2008). These accounts neglect the role of context in identifying and legitimising governance practices. Most importantly, the current accounts neglect the significance of being sensitive to the constant changes that occur in the context. I extend these views by demonstrating that the processes of analysis and design determine the governance practices that maintain the identity of the community based on the context.

One of the significant research findings is illustrating that contextualising the practices is performed through analysis and design, which are typical processes in the software development life cycle. The significance of these processes in vertical OSS communities was previously highlighted in the relevant literature (e.g. Fitzgerald, 2006). In this thesis, I contribute to the current literature by providing empirical evidence, not only on the role of analysis and design in vertical OSS communities, but on how analysis and

design trigger boundary decisions and produce governance practices. I also extend the current views by arguing that analysis and design are not limited to vertical OSS communities. Instead, they are ongoing processes in any OSS community to situate and legitimise governance practices in a particular context. I highlight that the context assess the dynamic changes of the community, and thus sets and resets the objectives of the community.

Moreover, as previously mentioned, one of the main objectives of the research is to highlight the role of materiality in governing OSS communities. Scholars (e.g. Faraj and Azad, 2012) demonstrate that, in order to conceptualise materiality, context has to be taken into consideration. The research findings illustrate that both the community and the technology in-use are in flux. Therefore, context is introduced as a construct that is representing the conditions that trigger boundary decisions.

7.2.2. Control

Control sets the community structure, and determines the domain of authority and influence (Jarvenpaa and Lang, 2011). Control has emerged as the core construct of the proposed governance model. This resonates with the current literature (e.g. West, 2003; Markus, 2007; O'Mahony and Ferraro, 2007) that has focused on controlling the behaviour of individuals and controlling the code development process as major practices to govern the community. It is evident from the relevant literature that control is the main motive to form OSS communities. The formation of Linux as the first OSS community was to allow users to control the destiny of their platforms (West, 2003). Besides, firms form alliances with OSS communities to retain control over the community resources to achieve their business success (West and O'Mahony, 2008). This suggests that controlling the resources is considered as the ultimate objective of governance and ensures the sustainability of the community.

Scholars argue that control is not directly imposed on the community (Dahlander and Magnusson, 2008). It is applied through various practices and mechanisms depending on the context of the community. In certain contexts, controlling resources is practiced through membership, licences, and voting procedures (Sharma et al., 2002), while in other context leadership was salient in controlling resources (Schaarschmidt et al., 2015). Moreover, informal control (i.e. social practices) practices are more common in autonomous communities; while sponsored communities experience more formal control. Scholars demonstrate that "heavy-handed" control mechanisms hinder participations in volunteer-based communities. On the other hand, domain-specific or mission-specific communities are in favour of strict control (Shah, 2006).

This suggests that control determines the other governance practices that best govern the community at that particular context. However, the current literature explains control by either describing the community structure, or the governance system. The community structure determines how decision rights and authority are distributed among different community members (Demil and Lecocq, 2006). As illustrated in the literature review, OSS structure have been described as network, bazaar, and peer-production. Each structure follow certain mechanism to efficiently exchange resources among contributors. The governance system refers to the rules that enable autonomous community members to share a basis of authority, such as bureaucratic, autocratic, and meritocratic forms of governance (O'Mahony and Ferraro, 2007).

Although the existing literature explained control as a governance mechanism, the antecedents and consequences of control needs further attention (Tiwana et al., 2013). In this thesis, I propose a governance model that departs from the current views by explaining control as an evolving and contextual governance practice. From a boundary decision perspective, I illustrate that control is about identifying the required

resources, and channelling them among the contributors in order to meet the objectives that were set by the processes of analysis and design.

7.2.3. Resources

As aforementioned, the ultimate objective of governance is to control the communal resources in order to produce an OSS product. In previous chapters I explained that resources refer to skills, ideas, money, time, OSS code, and technical artefacts. Early publications related to OSS governance (e.g. Benkler and Nissenbaum, 2006) describe OSS communities as a new form of organisation that convert dispersed knowledge into resources that are inaccessible through conventional forms of organisation. Therefore, firms tend to benefit from OSS communities to extend their resource base and accelerate their technological development (Dahlander and Magnusson, 2008). Similarly, OSS communities form alliances with firms to benefit from the resources that firm could provide (O'Mahony and Bechky, 2008).

It is evident that the literature is dominated by studies that focus on technical capabilities as the main community resource (Aksulu and Wade, 2010). This directs the attention towards the roles and responsibilities as a major property of human resources (Crowston et al., 2012). This explains why governance literature payed extensive amount of attention on attracting and retaining technical capabilities to sustain the community.

More recent studies demonstrate that the OSS communities are fluid and dynamic, and thus resources are in flux (Faraj et al., 2011; Aaltonen and Lanzara, 2015). This suggests that it is not possible to realise the potential resources and the roles at the outset. Moreover, the fluidity of the OSS community and the emergence of variety of resources produced divergent, and often, conflicting interests. Therefore, the topic of managing resources in the literature is often related to resolving tensions, mainly between the communities and firms (e.g. O'Mahony and Bechky, 2008), donators and rent seekers (i.e. hobbyists and professionals) (e.g. Franck and Jungwirth,

2003), and community openness and ownership (West and O'Mahony, 2008). In general, relevant literature focuses on the governance mechanism that takes the community to state of stability.

The proposed OSS governance model extends these views by acknowledging that stability is not a desirable state in OSS communities. This resonates with the more general argument by Faraj et al. (2011), who argue that further studies need to focus on how resources responds to the various tensions rather than examining the stable structural mechanisms of online communities. Therefore, the proposed OSS governance model focuses on the properties of resources rather than resources as whole. I illustrate that the properties are not fixed. Instead they emerge through interactions with the surrounding context. Therefore, resource, as a construct in the proposed model, represents the fluidity of the community.

7.2.4. Materiality

Given the significance of technology in OSS communities, their role in governance cannot be ignored. However, technology is considered in the relevant literature as separable from the surrounding context. The current literature explains OSS governance either using technological determinist or human-centred approaches. However, today, OSS communities represent complex ecosystems (Kallinikos et al., 2013). Therefore, there is a growing need for conceptualising the role of technology in the change and stability of the ecosystems (Tilson et al., 2010). However, technology is not fixed.

The research findings resonates with the existing literature in demonstrating that technology is editable, interactive, and generative. Generativity means that the technology is self-contained and capable of producing outcomes without human intervention (Tilson et al., 2010). It also refers to the unintended consequences of technology (Kallinikos et al., 2013). This suggests that technologies are prone to changes. Current studies (e.g. Chua and Yeow, 2010) urged researchers to conduct further

studies that surface the role of the material properties of technology in governing OSS communities.

As mentioned in chapter 2, the role of materiality in governing OSS communities in an under-theorised area of research. Therefore, I referred to a broader literature to understand how materiality is theorised. It is evident that the concept of materiality is debatable causing inconsistencies in the relevant literature (Jones, 2014). In this thesis, I have adhered to the definition of materiality that is grounded in the concept of affordance, which is rooted in a relational ontology that gives equal play to the material as well as the social aspect of the target phenomenon (Faraj and Azad, 2012).

Therefore, in the emergent governance model, materiality is introduced as a construct to stress its integral part of the organisational change (Orlikowski, 2007). The construct of materiality focuses on what the technology can perform in a given context and what are the consequences (Pentland and Singh, 2012). This highlights the specific properties of technologies that make a difference in boundary decisions, and thus in governing OSS communities.

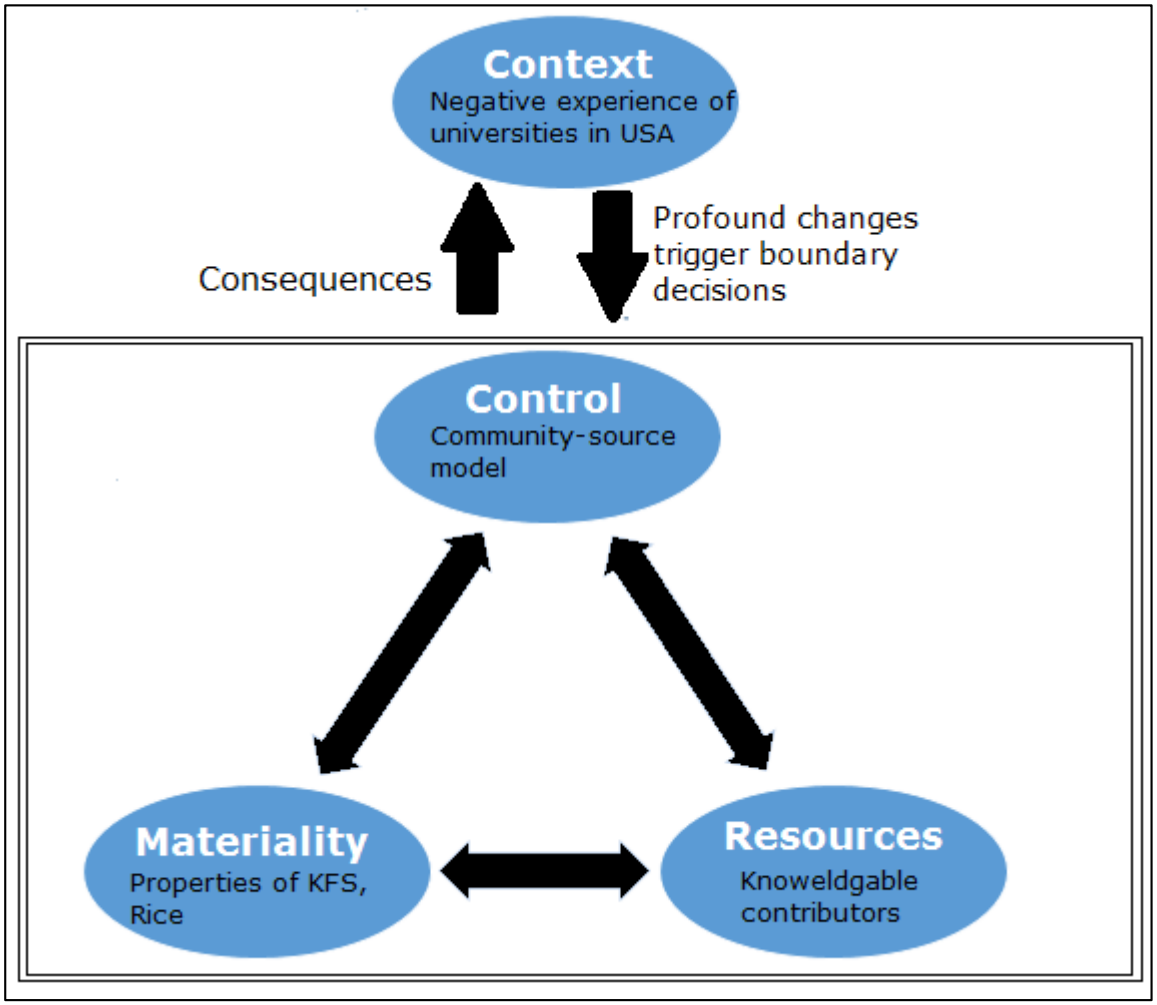
7.3. The Emergent Governance Model in Relation to Kuali

In this section, I illustrate the OSS governance model with relation to Kuali community. I explain how the constructs of context, control, resources, and materiality interacted with each other to produce boundary decisions that have governed Kuali community.

As previously mentioned, the construct of context represents the conditions that trigger boundary decisions. The conditions are determined by the ongoing processes of analysis and design. As illustrated in figure 21, the context prior to the inception of Kuali reflects the negative experiences of the universities in USA with regards to the acquisition and implementations

of ERP systems. This triggered the need for forming a university-specific, i.e. vertical, OSS community.

Figure 21: Creating the Community



Accordingly, creating the community was a boundary decision to delineate the boundaries of the community by determining who is included in the community, and what their rights are. It was also a boundary decision to delineate the software development process to ensure the quality and reliability of the OSS. This was essential to distinguish Kuali from other OSS communities and from competing ERP vendors.

As previously discussed, control is the ultimate objective of OSS governance. According to the governance model, control is practiced through the interaction between resources and materiality. As shown in figure 21, resources during the first governance phase were mainly about

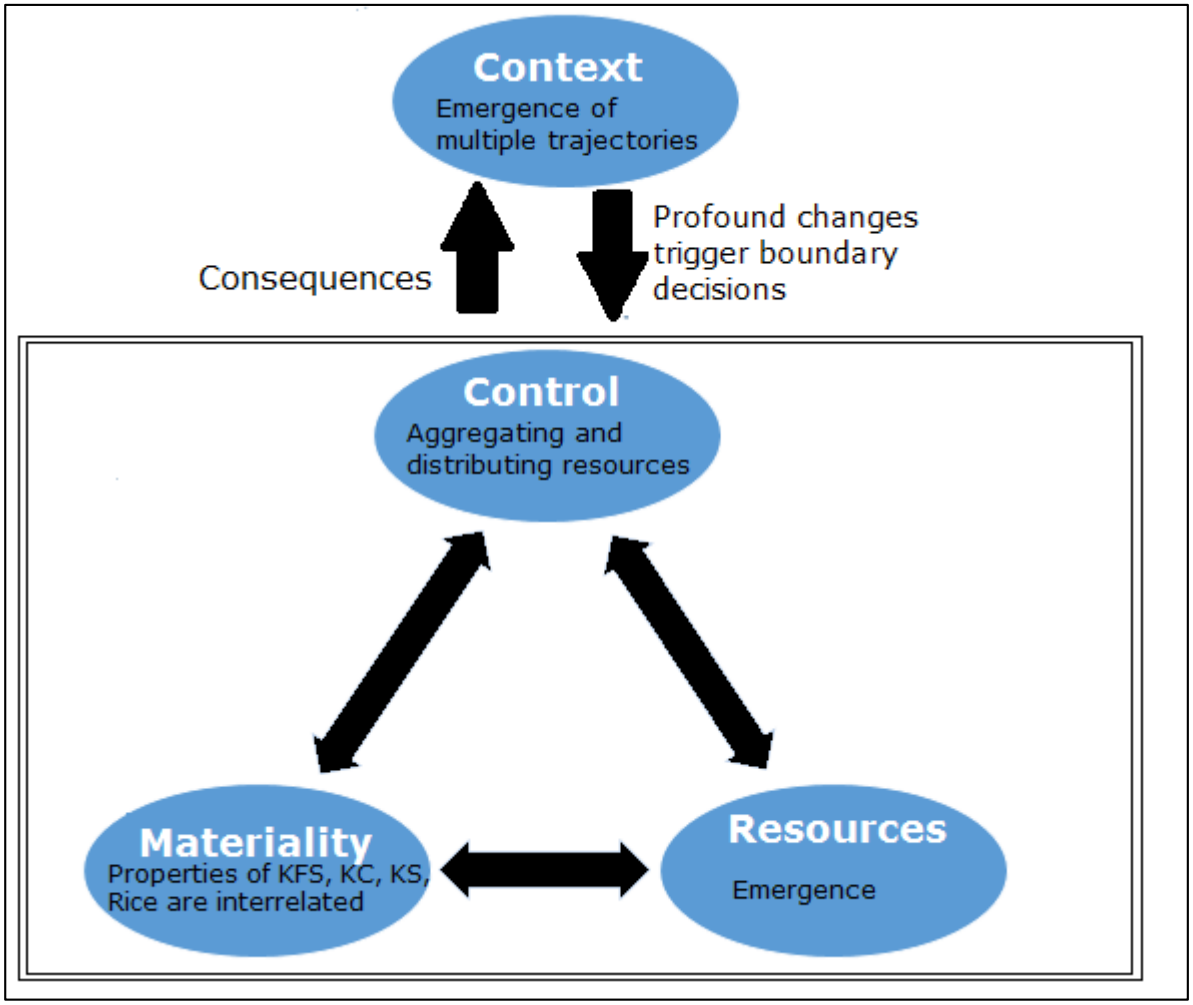
selecting knowledgeable contributors as the main resources to maintain a university-specific community. In terms of materiality, control was imposed mainly through, first, building KFS on top of the approved functionalities of IU's legacy finance system to attract potential universities. Second, control was imposed through imposing Rice framework to protect the software development process. This particular arrangement of the material properties of the KFS and Rice created an exclusive vertical OSS community that provides HE-specific finance system. Consequently, it attracted the interests of a wider range of stakeholders, and thus led to the expansion of the community.

Expanding the scope of Kuali increased the fluidity of the community, and thus resources were in flux. The main contextual factor that triggered the second boundary decision was the emergence of multiple trajectories, which gave rise to divergent interests. Therefore, balancing interests was a boundary decision that emerged to balance between maintaining diversity to enrich the community on one hand, and controlling the resources to maintain the quality and reliability of the OSS products on the other hand, as summarised in figure 22.

In general, deciding on the control mechanisms in OSS communities is challenging (Curto-Millet and Shaikh, 2017), especially with the existence of diversity (Schaarschmidt et al., 2015). The main challenge is how to set the boundaries of the community in a way that balances between control and growth. This is explained by Ferraro and O'Mahony (2012) as follows:

> "*If project boundaries remain open, how do they ensure that incoming contributors do not violate the project's mission? If project boundaries close, who become the gatekeepers of such forms?*" (p:545)

Figure 22: Balancing the Interests

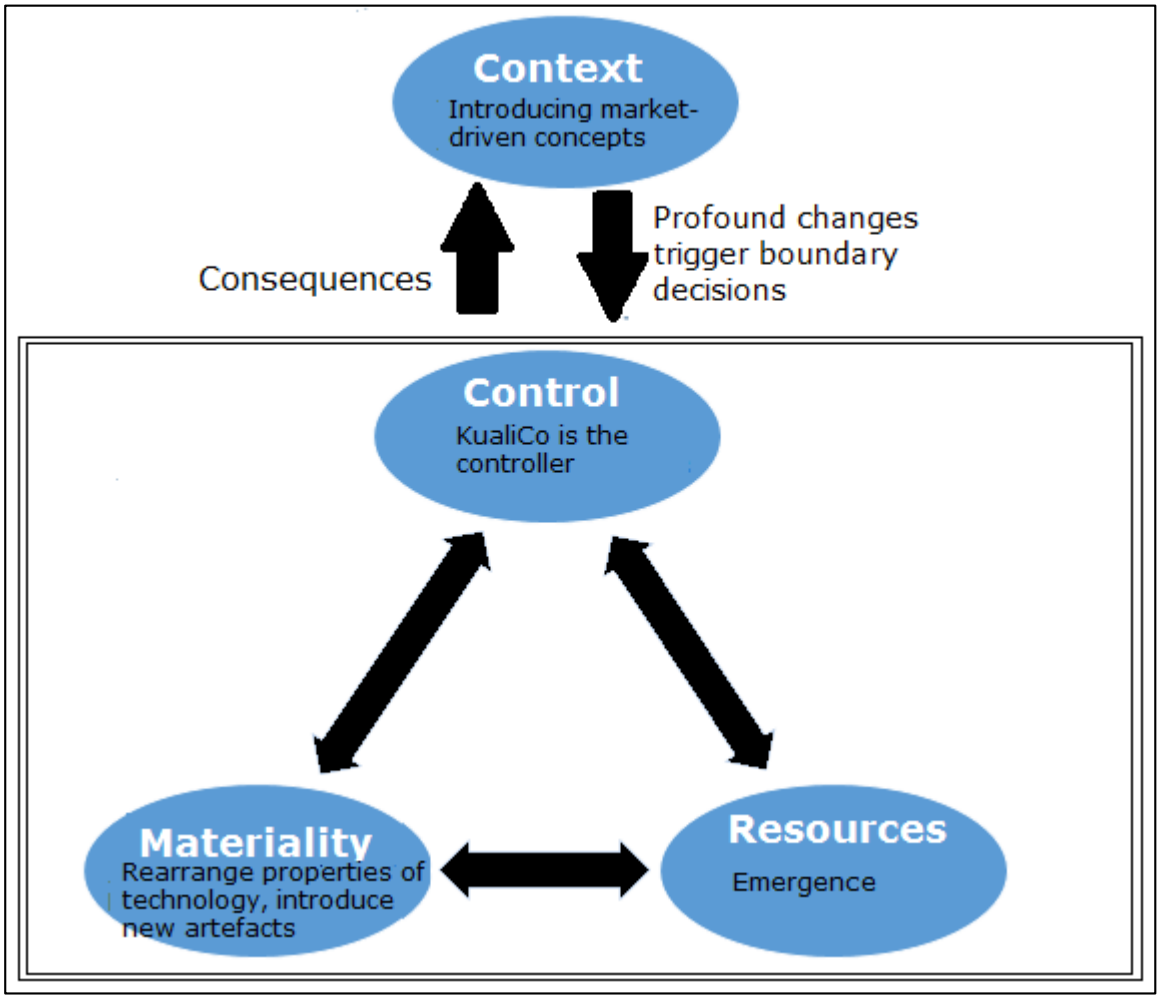


The governance practices that emerged are focused on aggregating and distributing resources. This is consistent with the current literature, which demonstrate that the diverse resources improve the quality of the product (O'Mahony and Bechky, 2008). However, resources are not fixed. Therefore, the salient governance practices during the second governance phase mainly focused on emergence by including the properties of resources, and excluding others. In terms of materiality, the properties of technology were constantly rearranged in relation to each other to accomplish governance practices.

Throughout the second governance phase, the resources and the collaborative efforts were directed towards fulfilling the requirements of universities. Besides, the technical and non-technical work practices were

conducted by university employees and constrained by university rules. This has risked the efficiency of Kuali projects. Thus, the Foundation introduced market-driven concepts to improve Kuali products, which has caused profound changes to the context of Kuali. This triggered the third boundary decision; sustaining the community.

Figure 23: Sustaining the Community



Sustaining the community is a boundary decision that aimed to improve Kuali products, and at the same time control the future direction of the community in a way that retains the university-specific nature of Kuali. The aim of this boundary decision was to reduce the risks of limiting contributions to the community of universities. Accordingly, the Foundation formed KualiCo.

Introducing KualiCo as the main player changed how the collective effort was controlled, as summarised in figure 23. This required reconfiguring the

structure of the community, and thus redefining the roles and responsibilities. The role of materiality was also evident during the third governance phase. KualiCo has changed the way artefacts were used. For example, KualiCo started using the internal cubes in Jira for their private discussions. This suggests that OSS communities do not solely rely on complete openness. Instead, they tend to balance openness based on the context. This is consistent with the findings of Shaikh and Vaast (2016), who illustrate that extreme openness in some cases can be limiting and lead to extreme failures. They demonstrate that OSS communities experience shifts in their setting, and accordingly they tend to experience moments of closure and opacity.

To sum up, Kuali was originally inspired by the ideology of the American universities during the early 2000's. However, how the community actually works today turns out to be very different from the original idea. This transformation in the context of Kuali community; i.e. from community source to market-driven community, is consistent with the findings of the current literature. In a recent study, Curto-Millet and Shaikh (2017) demonstrate that the continuous change in the boundaries ensures the sustainability of the community. They argue that reducing OSS to the opposite of the proprietary software "does not do justice to the community". This suggests that the community governs the collective effort by changing the boundaries based on the change in the context; i.e. boundary decisions.

7.4. The Emergent Governance Model in Relation to the Existing Literature

In the relevant literature, boundary decisions in online communities in general are discussed using various constructs that tend to describe boundaries as a static demarcation that separates individuals and organisations (Akoumianakis, 2014). They are explained in terms of spatial and temporal constructs to explain the changes that occur to the

boundaries of the community (Watson-Manheim et al., 2012). Besides, demographic dimensions are also considered as a boundary construct in research that focuses in social distance (Gibson and Gibbs, 2006). In addition, it is evident in the literature that boundary decisions are also discussed in terms of boundary objects (e.g. O'Mahony and Bechky, 2008), which are any sort of arrangements that facilitates collaboration (Star, 2010). The OSS communities tend to form an organisation, either for-profit or not-for-profit, to act as a boundary object that negotiates the divergent interests of the community members (O'Mahony and Bechky, 2008).

These explanations were acceptable in explaining the settings of autonomous communities that are self-governed and rely on voluntarily efforts. However, these constructs, such as time and space, do not provide sufficient explanations on the specificity of the governance practices (Howison and Crowston, 2014). Besides, the advancements in technology fragmented resources across multiple functional, geographical, hierarchical, and professional boundaries, which requires multi-dimensional constructs (Lindgren et al., 2008). Therefore, recent literature has developed broader views of boundaries to provide a deeper understanding of boundary decisions in online communities. One of the explanations that has attracted the interest of scholars is the explanation provided by Santos and Eisenhardt (2005).

Santos and Eisenhardt (2005) argue that boundaries reflect the essence of the organisation. Therefore, they propose four boundary constructs that have been given a great attention in the literature of OSS and online communities because they provide a dynamic understanding of the changes that occur to the boundaries of the community, and thus unfold the emergence and dynamicity of the governance practices (Liang et al., 2016). These constructs are: power, identity, competence, and transactional efficiency. Power is about controlling the resources, reducing uncertainty, improving the performance, and managing relationship with external entities (Liang et al., 2016). Identity is concerned about defining "who we

are". Competence is about developing the capabilities of the community members; especially with regards to the administrative and technical skills (Jarvenpaa and Lang, 2011; Teigland et al., 2014). Transactional efficiency focuses on facilitating collaboration and providing the community with artefacts that develop the OSS product (Jarvenpaa and Lang, 2011).

In the context of online communities in general, there are modest attempts to explain boundary decisions using the constructs proposed by Santos and Eisenhardt. One of the prominent studies is the one conducted by Jarvenpaa and Lang (2011). They have adopted a holistic perspective to explain how online communities change their boundaries overtime. Based on a comparison between sponsored and autonomous communities, they illustrate the interactions and interdependencies among power, identity, competence, and efficiency to balance between openness and control. While they have underlined the importance of context in defining how the community controls the resources, their study did not touch upon the issue of materiality.

Besides the work of Santos and Eisenhardt, there are other modest attempts to highlight the multi-dimensional nature of community boundaries. For example, the research conducted by Chua and Yeow (2010) takes into consideration the dynamicity and the materiality of the coordination practices in an OSS gaming community. They provide empirical evidence to explain coordination practices in terms of actors, interactions, and artefacts. Another example is the research conducted by Aaltonen and Lanzara (2015). They have studied Wikipedia community and observed the changes that has occurred to the community over a period of 9 years. They illustrate how Wikipedia governed the collectives from inception to maturity. They have demonstrated OSS governance as an evolving, enabling, and embedded process. Evolving means that the community changes its practices as the context changes. Enabling refers to the materiality of the technology that enable governance practices. Embedded means that governance does not reside on an administrative

body or performed through defined functional routines. Instead, OSS governance is implicit in the technology and in the interactions between the actors. However, they have based their study on the assumption that OSS communities lack contracts and authority to control the behaviours of individuals, which neglects the fact that formal organisations coexist with OSS communities adding sort of formalities to OSS governance practices.

The proposed OSS governance model relates to these influential explanations. However it differs from them by proposing a more comprehensive theoretical foundation that does not overemphasise on human actors or non-human actors. Besides, the proposed OSS governance model balances between the openness and control to maintain the standards of the community. The emergent OSS governance model contributes to the theoretical understanding of how OSS communities govern the collectives through boundary decisions. Although this model is grounded on a particular context; i.e. vertical OSS community, it has implications to a wider range of phenomena in the field of information systems and organisation studies.

7.5. Research Evaluation Criteria

In this section I assess the validity and credibility of the thesis. This is achieved by evaluating the emergent grounded theory based on the criteria proposed by Corbin and Strauss (1990).

Criterion 1: Are concepts generated?

As clarified in chapter 3, the open coding phase began with conceptualisation. The concepts emerged from the empirical data using the participants' words in case of primary data, and using words that are close to data in terms of secondary data.

Criterion 2: Are the concepts systematically related?

After conceptualising the raw data, the concepts were compared to each other in order to create categories. This involved sorting, synthesising, and aggregating the concepts into categories that relate to the research inquiry.

Criterion 3: Are the categories well developed? Do categories have conceptual density or theoretical saturation?

I have adopted the Straussian coding paradigm that assisted in systematically generating categories, sets their properties, and extensively comparing and contrasting them to data. Besides, I have verified the conceptual density, i.e. theoretical saturation, by evaluating whether the emergent substantive theory explained the case under study. I have also illustrated that the emergent theory provided insights for a wider context, as will be further explained in chapter 8.

Criterion 4: Are the broader conditions that affect the phenomenon under study built into its explanation?

The proposed OSS governance model was based on a study on Kuali community during the period between 2004 and 2016. Kuali community went through various complexities that represent a variation of conditions, interactions, and consequences. Although Kuali represents a single OSS community, it comprises of multiple OSS projects that have their own specific contexts, objectives, user groups, resources, and consequences. Therefore, the proposed OSS governance model takes into consideration wide range of OSS contexts. In addition, it shows relevance to the governance of online communities in general.

Criterion 5: Has process been taken into account?

Process here refers to the temporal dimension of the phenomenon under study. In this thesis, I describe the process of movement of Kuali community since its inception. This was described in terms of governance phases that emerged from conditions, experienced various interactions, and produced consequences.

Criterion 6: Do the theoretical findings seem significant and to what extent?

The research findings and contributions are salient. As will be explained in chapter 8, the research findings contribute to the literatures of OSS governance, and boundary management. Besides, the findings have practical contributions. In addition, the proposed OSS governance model stimulates further studies.

Chapter 8: Conclusion and Thesis Contributions

8.1. Introduction

I begin this chapter by concluding the thesis. Then, I illustrate the potential contributions of the research. Finally, I summarise the limitations of the research, and I provide insights for future studies.

8.2. Conclusion: Answering the Research Questions

This thesis started with a general objective to understand how OSS governance practices emerge and evolve overtime. During data collection and analysis, two main aspects of governance have emerged. First, OSS communities govern their collectives through changing their boundaries; i.e. boundary decisions. Second, OSS governance cannot be sufficiently explained without encountering materiality.

This PhD contributes to the existing literature by developing an OSS governance model to explain OSS governance. The way I have approached OSS governance departs from the dominant perspectives in the relevant literature. Although, I resonate with the current accounts in describing OSS governance as emergent and evolving overtime, I depart from them in bringing to the fore the dynamicity and materiality aspects of governance. The proposed OSS governance model highlights the importance of shifting the current emphasis of OSS literature to the hybrid forms of OSS communities that merge OSS ethos with the market-oriented software development practices.

This thesis has achieved its objectives by developing a substantive theory to explain OSS governance. The substantive theory is represented through an OSS governance model. The governance model illustrates that OSS communities are governed through boundary decisions. These decisions delineate the boundaries of the community through identifying the actors, resources, and actions required to control and coordinate the collective effort in a particular context.

The answer to the first research question, *how governance practices emerge in vertical OSS communities?* Based on the OSS governance model, governance practices emerge from the processes of analysis and design that are necessary to contextualise and legitimise the governance practices. In other words, the context forms the antecedent conditions for governance practices. This challenges the predominant assumptions that the OSS governance is determined by the communication patterns of the developers. It was evident that analysis and design were ongoing throughout Kuali life. The community continuously observes the changes that occur to Kuali projects and evaluates the existing governance practices.

The relevant literature raised open questions regarding how frequent the community members need to observe the changes in the community and react accordingly? (Singh and Tan, 2010), and who is responsible for analysis and design? (Fitzgerald, 2006). In this thesis, I have illustrated that these processes are ongoing. However, boundary decisions are taken only after the occurrence of profound changes on the context and objectives of the community. Regarding who is responsible for analysis and design, Fitzgerald (2006) argue that in horizontal OSS communities all development processes (planning, analysis, design, and implementation) are performed by the same developer, while in vertical communities they are performed by different knowledgeable developers.

My research findings support this argument as the findings demonstrate that analysis and design were performed by different knowledgeable individuals; not necessary developers, during different governance phases. During the first governance phase, university representatives were observing the dynamic changes and performing analysis and design. During the second governance phase, this responsibility was assigned to the Board members in Kuali Foundation. Then KualiCo formed a dedicated team to handle the analysis and design.

With regards to second research question, *how OSS community governing contribution shapes its boundaries over time*? I have demonstrated in section 7.3 that OSS communities govern their collective effort through boundary decisions, which I have explained through the interaction between the constructs of context, control, resources, and materiality.

8.3. Thesis Contributions

Theorising OSS governance was inspired by the empirical data. However, it has implications beyond the case of Kuali, as summarised in table 17. This thesis represents a unique form of OSS community that challenges the taken-for-granted assumptions on OSS governance. In addition, the thesis demonstrates the governance of a community from inception to maturity. This is a contribution to the OSS literature that often neglects the initial stages of the community and assumes that the community is in a mature stable state and needs governance to be sustained (Crowston et al., 2012; Hallerbach et al., 2013).

Area	**Contribution**
OSS Governance	-Explain OSS governance beyond the dyadic firm-community relation -Theorise OSS governance as ongoing boundary decisions -Highlight the role of materiality in governing OSS communities
Boundary Management	-Move beyond the concept of efficiency in explaining boundary decisions -Adopt problem-driven boundary phenomena
Practical Contributions	-Highlight the importance of analysis and design in contextualising governance practices -Recognise that the requirements of the community changes as the community evolves

Table 17: Summary of Thesis Contribution

The thesis also contributes to the literature on OSS governance by providing empirical and theoretical insights that move beyond the conventional dyadic relation between the firm and the OSS community. The existing literature is dominated by explanations of OSS governance where

OSS is a software development strategy in the firm (e.g. Rolandsson et al., 2011). This explains the dynamicity and stability of the community as the result of tensions between the community and the firm, which does not adequately represents the current status of the OSS communities. There are growing concerns in the literature of technology and organisation studies in general (e.g. Tilson et al., 2010; Tiwana et al., 2010) with regards to the dynamicity and stability of modern digital infrastructures and ecosystems, such as OSS communities. This thesis proposes a governance model that stresses the importance of emergence, which enables complex and dynamic interplay between variety of resources and stakeholders. The proposed model also focuses on the consequences of the interplay. This provides a deeper understanding of OSS governance beyond community-firm relationship.

Besides, introducing materiality as a theoretical construct is an important theoretical contribution because it addresses critical, yet poorly understood, issues with regards to stability and dynamicity. For example, the thesis responds to the recent calls for understanding the consequences of technology in complex ecosystems (e.g. Kallinikos et al., 2013; Eck and Uebernickel, 2016). The thesis demonstrates that technology is generative, where generativity is considered as a powerful concept in describing contemporary transformations that has occurred to the OSS communities. Therefore, proposing a theoretical model that surfaces the materiality of technology is a major contribution (Yoo, 2013).

The thesis also contributes to the studies related to boundary management in online communities. Boundary decisions have strategic importance in any organisation (Poppo and Zenger, 1998). Santos and Eisenhardt (2005) urged researchers to advance the literature of boundary management through adopting integrative view of the various boundary conceptions. They urged scholars to move beyond the concept of efficiency in explaining boundary decisions. On way of doing so, according to Santos and Eisenhardt (2005), is to focus on boundary decisions in environments that

do not solely rely on efficiency. As explained in chapter 2, OSS communities represent a unique form of organisation that cannot be explained through market and hierarchal governance mechanisms. The research findings contribute to the literature of boundary management by providing empirical evidence on the role of non-efficiency boundary concepts in shaping boundary decisions. This thesis reviews the traditional assumptions of boundary decisions and contributes to the development of new theoretical concepts.

In addition, Santos and Eisenhardt (2005) also called for more research that focus on problem-driven boundary phenomena. The literature on boundary decisions in often theory-driven. These type of studies neglect the contemporary organisations that have been transformed by boundary decisions. Therefore, adopting a problem-driven research may lead to novel considerations and innovative theoretical insights. In this thesis, I have adopted a grounded theory approach that has started from the data, where the case represents non-traditional organisational settings. Besides, the emergent governance model highlights the role of contextualisation. This responds to the call by Puranam et al. (2014) for developing theory that explains and predicts emergent forms of organising.

Besides the theoretical contributions, the thesis also have important practical implications, especially in light of the proliferation of OSS communities in various industries (Haefliger et al., 2011). The proposed OSS governance model can be useful for practitioners in terms of structuring the OSS communities. The research findings demonstrate that the governance of OSS communities is inseparable from its context. Therefore, OSS governance is influenced by the purpose of the OSS product, the type of the software, and the type of stakeholders. Some communities are sponsored by firms, and thus focus on leveraging innovation. Others are more focused on the public good, and accordingly focusing on how to attract and sustain participation.

The proposed OSS governance model encourages practitioners to be aware of the specificity of the target OSS product. This also emphasises the importance of avoiding the act of imposing formalised structures and roles to achieve stability. The research findings illustrate that it is possible to achieve stability while allowing resources and roles to emerge based on the context. Besides, with respect to materiality, the research findings illustrate that the materialisation of technologies make a difference in OSS governance. This draws the attention to the design of the OSS code and the artefacts in use and their significant implications on governance. The design should be flexible to accommodate the emergence of the OSS community.

8.4. Limitations and Suggestions for Future Work

The theory and empirical work presented in this thesis makes useful and significant contributions, albeit not without limitations. First, the context of country was limited to USA, South Africa, and Canada. Universities from Europe were meant to be included to enrich the research; however such universities were difficult to reach due to the poor online documentations with regards to their collaboration in Kuali community.

Second, there are other aspects of materiality that have not been covered in the research. After the formation of KualiCo, social media was introduced as an essential artefact for communication and collaboration, which may provide more variations in validating the proposed OSS governance model. Third, a single case study is useful in the case of Kuali, which is a novel case. However, additional case studies can be further conducted to test the emergent substantive theory.

In terms of future research, the thesis offers a theoretical foundation for other scholars to validate and refine. The emergent grounded theory can be further validated and compared within existing theories to promote a more formal theory to explain OSS governance. The findings of the thesis also provide insights for developing the methodological tool used to explore

OSS communities. Highlighting the role of materiality encourages further research to broaden the scope of acceptable and credible data sources. One way of doing so is digital traceability. The technical artefacts produce massive amount of traceable data that represent the interactions in the community. Such data can supplement primary and archived data, which are the most common data sources in studies related to OSS. Digital traces are considered as a promising data sources for theorising in the field of organisation studies and information systems.

References

Aaltonen, A. and Lanzara, G. F. 2015. "Building Governance Capability in Online Social Production: Insights from Wikipedia". *Organization Studies,* 36 (12)**,** 1649-1673.

Adler, P. S. and Heckscher, C. 2006. Towards collaborative community. *In:* Adler, P. S. & Heckscher, C. (eds.) *The firm as a collaborative community: Reconstructing trust in the knowledge economy.* New York: Oxford University Press.

Agerfalk, P. J. and Fitzgerald, B. 2008. "Outsourcing to an unknown workforce: Exploring opensourcing as a global sourcing strategy". *MIS Quarterly,* 32 (2)**,** 385-409.

Akoumianakis, D. 2014. "Boundary Spanning Tactics and "Traceable" Connections in Cross-Organizational Virtual Alliances: A Case Study". *Journal of Enterprise Information Management,* 27 (2)**,** 197-227.

Aksulu, A. and Wade, M. 2010. "A Comprehensive Review and Synthesis of Open Source Research". *Journal of the Association for Information Systems,* 11 (11)**,** 576-656.

Ancona, D. G. and Caldwell, D. 1990. "Beyond Boundary Spanning: Managing External Dependence in Product Development Teams". *The Journal of High Technology Management Research,* 1 (2)**,** 119-135.

Andreev, P., Feller, J., Finnegan, P. and Moretz, J. Conceptualizing the Commons-Based Peer-production of Software: an Activity Theoretic Analysis. ICIS 2010 Proceedings, 2010. 43.

Arazy, O., Daxenberger, J., Lifshitz-Assaf, H., Nov, O. and Gurevych, I. 2016. "Turbulent Stability of Emergent Roles: The Dualistic Nature of Self-Organizing Knowledge Coproduction". *Information Systems Research,* 27 (4)**,** 792-812.

Baldwin, C. Y. and Clark, K. B. 2006. "The Architecture of Participation: Does Code Architecture Mitigate Free Riding in the Open Source Development Model?". *Management Science,* 52 (7)**,** 1116-1127.

Barcellini, F., Détienne, F. and Burkhardt, J. M. 2008. "User and Developer Mediation in an Open Source Software Community: Boundary Spanning through Cross Participation in Online Discussions". *International Journal of Human-Computer Studies,* 66 (7)**,** 558-570.

Barney, J. B. 1999. "How a Firm's Capabilities Affect Boundary Decisions". *Sloan Management Review,* 40 (3)**,** 137.

Barrett, M., Heracleous, L. and Walsham, G. 2013. "A Rhetorical Approach to IT Diffusion: Reconceptualizing the Ideology-Framing Relationship in Computerization Movements". *MIS Quarterly,* 37 (1).

Barrett, M., Oborn, E. and Orlikowski, W. 2016. "Creating Value in Online Communities: The Sociomaterial Configuring of Strategy, Platform, and Stakeholder Engagement". *Information Systems Research,* 27 (4)**,** 704-723.

Benkler, Y. 2002. "Coase's Penguin, or Linux and The Nature of the Firm". *Yale Law Journal,* 112 (3)**,** 369-447.

Benkler, Y. and Nissenbaum, H. 2006. "Commons-based Peer Production and Virtue*". *Journal of Political Philosophy,* 14 (4)**,** 394-419.

Benlian, A. and Hess, T. 2011. "Comparing the Relative Importance of Evaluation Criteria in Proprietary and Open-Source Enterprise Application Software Selection–A Conjoint Study of ERP and Office Systems". *Information Systems Journal,* 21 (6)**,** 503-525.

Bird, C., Pattison, D., D'Souza, R., Filkov, V. and Devanbu, P. Latent Social Structure in Open Source Projects. Proceedings of the 16th ACM SIGSOFT International Symposium on Foundations of software engineering, 2008. ACM, 24-35.

Bolici, F., Howison, J. and Crowston, K. 2016. "Stigmergic Coordination in FLOSS Development Teams: Integrating Explicit and Implicit Mechanisms". *Cognitive Systems Research,* 38**,** 14-22.

Bonaccorsi, A. and Rossi, C. 2003. "Why Open Source Software Can Succeed". *Research Policy,* 32**,** 1243–1258.

Boudreau, M.-C. and Robey, D. 2005. "Enacting Integrated Information Technology: A Human Agency Perspective". *Organization Science,* 16 (1)**,** 3-18.

Brooks, L. 2007. Considering Open Source: A Framework for Evaluating Software in the New Economy. *EDUCAUSE Center for Analysis and Research (ECAR)* [Online]. Available: http://www.educause.edu/library/resources/considering-open-source-framework-evaluating-software-new-economy [Accessed 05/06/2014].

Charmaz, K. 2014. *Constructing Grounded Theory*, Sage.

Chen, K. K. and O'Mahony, S. 2009. "Differentiating Organizational Boundaries". *Research in the Sociology of Organizations,* 26**,** 183-220.

Chua, C. E. H. and Yeow, A. Y. K. 2010. "Artifacts, actors, and interactions in the cross-project coordination practices of open-source

communities". *Journal of the Association for Information Systems,* 11 (12)**,** 838-867.

Coase, R. H. 1937. "The Nature of the Firm". *Economica,* 4 (16)**,** 386-405.

Corbin, J. and Strauss, A. 1990. "Grounded Theory Research: Procedures, Canons and Evaluative Criteria". *Qualitative Sociology,* 13 (1)**,** 418-427.

Corbin, J. and Strauss, A. 2008. *Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory*, Thousand Oaks.

Cornford, T., Shaikh, M. and Ciborra, C. 2010. "Hierarchy, laboratory and collective: Unveiling Linux as innovation, machination and constitution". *Journal of the Association for Information Systems,* 11 (12)**,** 809-837.

Courant, P. N. and Griffiths, R. J. 2006. Software and collaboration in higher education: A study of open source software. Available: http://www.campussource.de/opensource/docs/OOSS_Report.pdf [Accessed 30/04/2015].

Creswell, J. W. 2007. *Qualitative Inquiry and Research Design: Choosing Among Five Approaches*, Sage Publication.

Crowston, K., Li, Q., Wei, K., Eseryel, U. Y. and Howison, J. 2007. "Self-organization of Teams for Free/Libre Open Source Software Development". *Information and Software Technology,* 49 (6)**,** 564-575.

Crowston, K., Wei, K., Howison, J. and Wiggins, A. 2012. "Free/Libre Open-Source Software Development: What We Know and What We Do Not Know". *ACM Computing Surveys,* 44 (2).

Curto-Millet, D. and Shaikh, M. 2017. "The Emergence of Openness in Open-Source Projects: The Case of OpenEHR". *Journal of Information Technology***,** 1-19.

Dahlander, L. and Magnusson, M. 2005. "Relationships between Open Source Software Companies and Communities: Observations from Nordic Firms". *Research Policy,* 34 (4)**,** 481-493.

Dahlander, L. and Magnusson, M. 2008. "How Do Firms Make Use Of Open Source Communities?". *Long Range Planning,* 41 (6)**,** 629-649.

de Laat, P. B. 2007. "Governance of Open Source Software: State of the Art". *Journal of Management & Governance,* 11 (2)**,** 165-177.

De Noni, I., Ganzaroli, A. and Orsi, L. 2013. "The Evolution of OSS Governance: A Dimensional Comparative Analysis". *Scandinavian Journal of Management,* 29 (3)**,** 247-263.

De Paoli, S. and D'Andrea, V. 2008. "How Artifacts Rule Web-Based Communities: Practices of Free Software Development". *International Journal of Web Based Communities,* 4 (2)**,** 199-219.

Demil, B. and Lecocq, X. 2006. "Neither Market nor Hierarchy nor Network: The Emergence of Bazaar Governance". *Organization Studies,* 27 (10)**,** 1447-1466.

Deodhar, S. J., Saxena, K. B. C., Gupta, R. K. and Ruohonen, M. 2012. "Strategies For Software-Based Hybrid Business Models". *The Journal of Strategic Information Systems,* 21 (4)**,** 274-294.

Di Tullio, D. and Staples, D. S. 2013. "The Governance and Control of Open Source Software Projects". *Journal of Management Information Systems,* 30 (3)**,** 49-80.

Eck, A. and Uebernickel, F. Reconstructing Open Source Software Ecosystems: Finding Structure in Digital Traces. Thirty Seventh International Conference on Information Systems, 2016 Dublin.

Eisenhardt, K. M. 1989. "Building Theories from Case Study Research". *Academy of Management Review,* 14 (4)**,** 532-550.

Faraj, S. and Azad, B. 2012. The Materiality of Technology: An affordance Perspective. *In:* Leonardi, P. M., Nardi, B. A. & Kallinikos, J. (eds.) *Materiality and Organizing: Social Interaction in a Technological World.* Oxford University Press.

Faraj, S., Jarvenpaa, S. L. and Majchrzak, A. 2011. "Knowledge Collaboration in Online Communities". *Organization Science,* 22 (5)**,** 1224-1239.

Faraj, S., von Krogh, G., Monteiro, E. and Lakhani, K. R. 2016. "Special Section Introduction—Online Community as Space for Knowledge Flows". *Information Systems Research,* 27 (4)**,** 668-684.

Feller, J., Finnegan, P., Fitzgerald, B. and Hayes, J. 2008. "From Peer Production to Productization: A Study of Socially Enabled Business Exchanges in Open Source Service Networks". *Information Systems Research,* 19 (4)**,** 475-493.

Ferraro, F. and O'Mahony, S. 2012. Managing the Boundaries of an "Open" Project. *In:* Padgett, J. F. & Powell, W. W. (eds.) *The Emergence of Organizations and Markets.* Princeton University Press.

Fitzgerald, B. 2006. "The Transformation of Open Source Software". *MIS Quarterly,* 30 (3)**,** 587-598.

Franck, E. and Jungwirth, C. 2003. "Reconciling Rent-Seekers and Donators–The Governance Structure of Open Source". *Journal of Management & Governance,* 7 (4)**,** 401-421.

Gallivan, M. J. 2001. "Striking a Balance between Trust and Control in a Virtual Organization: A Content Analysis of Open Source Software Case Studies". *Information Systems Journal,* 11 (4)**,** 277-304.

Germonprez, M., Kendall, J. E., Kendall, K. E. and Young, B. 2014. "Collectivism, Creativity, Competition, and Control in Open Source Software Development: Reflections on the Emergent Governance of the SPDX® Working Group". *International Journal of Information Systems and Management,* 1 (1-2)**,** 125-145.

Gibson, C. B. and Gibbs, J. L. 2006. "Unpacking the Concept of Virtuality: The Sffects of Geographic Dispersion, Electronic Dependence, Dynamic Structure, and National Diversity on Team Innovation". *Administrative Science Quarterly,* 51 (3)**,** 451-495.

Gioia, D. A., Corley, K. G. and Hamilton, A. L. 2013. "Seeking Qualitative Rigor in Inductive Research: Notes on the Gioia Methodology". *Organizational Research Methods,* 16 (1)**,** 15-31.

Glaser, B. and Strauss, A. 1967. *The Discovery of Grounded Theory: Strategies for Qualitative Research*, Routledge.

Glaser, B. G. 2002. "Conceptualization: On Theory and Theorizing Using Grounded Theory". *International Journal of Qualitative Methods,* 1 (2)**,** 23-38.

Goulding, C. 2002. *Grounded Theory: A Practical Guide for Management, Business and Market Researchers*, Sage.

Grant, G. and Tan, F. B. 2013. "Governing IT in inter-organizational relationships: Issues and future research". *European Journal of Information Systems,* 22 (5)**,** 493-497.

Haefliger, S., Monteiro, E., Foray, D. and Von Krogh, G. 2011. "Social Software and Strategy". *Long Range Planning,* 44 (5)**,** 297-316.

Hallerbach, I. P., Barrett, M. and Faraj, S. 2013. Emergence in Nascent Online Communities: an Affordance Perspective. *Thirty Fourth International Conference on Information Systems.* Milan.

Hemetsberger, A. and Reinhardt, C. 2009. "Collective Development in Open-Source Communities: An Activity Theoretical Perspective on Successful Online Collaboration". *Organization Studies,* 30 (9)**,** 987-1008.

Holcomb, T. R. and Hitt, M. A. 2007. "Toward a Model of Strategic Outsourcing". *Journal of operations management,* 25 (2)**,** 464-481.

Howison, J. and Crowston, K. 2014. "Collaboration Through Open Superposition: A Theory of the Open Source Way". *MIS Quarterly,* 38 (1)**,** 29-50.

Izquierdo, J. L. C. and Cabot, J. Enabling the Definition and Enforcement of Governance Rules in Open Source Systems. Software Engineering (ICSE), 2015 IEEE/ACM 37th IEEE International Conference on, 2015. IEEE, 505-514.

Jarvenpaa, S. L. and Lang, K. R. 2011. "Boundary Management in Online Communities: Case Studies of the Nine Inch Nails and ccMixter Music Remix Sites". *Long Range Planning,* 44 (5)**,** 440-457.

Jeppesen, L. B. and Frederiksen, L. 2006. "Why Do Users Contribute to Firm-Hosted User Communities? The case of Computer-Controlled Music Instruments". *Organization Science,* 17 (1)**,** 45-63.

Jones, C., Hesterly, W. S. and Borgatti, S. P. 1997. "A General Theory of Network Governance: Exchange Conditions and Social Mechanisms". *Academy of Management Review,* 22 (4)**,** 911-945.

Jones, M. 2014. "A Matter of Life and Death: Exploring Conceptualizations of Sociomateriality in the Context of Critical Care". *MIS Quarterly,* 38 (3)**,** 895-925.

Kallinikos, J., Aaltonen, A. and Marton, A. 2013. "The Ambivalent Ontology of Digital Artifacts". *Mis Quarterly,* 37 (2).

Kane, G. C., Johnson, J. and Majchrzak, A. 2014. "Emergent Life Cycle: The tension between Knowledge Change and Knowledge Retention in Open Online Coproduction Communities". *Management Science,* 60 (12)**,** 3026-3048.

Kelle, U. 2010. The Development of Categories: Different Approaches in Grounded Theory. *In:* Bryant, A. & Charmaz, K. (eds.) *The Sage Handbook of Grounded Theory.* London: Sage.

Kuali Foundation. 2014. *Kuali Foundation* [Online]. Available: https://www.kuali.org/ [Accessed 25/03/2015].

Lakhani, K. R. and Wolf, R. G. 2005. Why Hackers do What they do: Understanding Motivation and Effort in Free/Open Source Software Projects. *In:* Feller, J., Fitzgerald, B., Hissam, S. & Lakhani, K. R. (eds.) *Perspectives on Free and Open Source Software.* MIT Press.

Lattemann, C. and Stieglitz, S. 2005. Framework for Governance in Open Source Communities. Proceedings of the 38th Annual Hawaii International Conference on System Sciences, 2005 Hawaii. IEEE.

Lauritzen, G. D., Salomo, S. and La Cour, A. 2013. "Dynamic Boundaries of User Communities: Exploiting Synergies Rather than Managing Dilemmas". *International Journal of Technology Management,* 63 (3-4)**,** 148-168.

Leonardi, P. M. 2011. "When Flexible Routines Meet Flexible Technologies: Affordance, Constraint, and the Imbrication of Human and Material Aagencies". *MIS Quarterly,* 35 (1)**,** 147-167.

Lerner, J. and Tirole, J. 2002. "Some Simple Economics of Open Source". *The Journal of Industrial Economics,* 50 (2)**,** 197-234.

Li, X., Yoo, Y. and Zhang, Z. 2016. Searching for "Stability" in Fluidity: A Routine-based View of Open Source Software Development Process. *Thirty Seventh International Conference on Information Systems.* Dublin.

Liang, Y., Zhou, R., Chen, J. and Wei, K. The Impact of Power Boundary Management on the Design of Company-Initiated Open Innovation Platform. PACIS, 2016.

Lindgren, R., Andersson, M. and Henfridsson, O. 2008. "Multi-Contextuality in Boundary-Spanning Practices". *Information Systems Journal,* 18 (6)**,** 641-661.

Ljungberg, J. 2000. "Open Source Movements as a Model for Organising". *European Journal of Information Systems,* 9 (4)**,** 208-216.

Locke, K. 2001. *Grounded Theory in Management Research*, Sage.

Markus, M. L. 2007. "The Governance Of Free/Open Source Software Projects: Monolithic, Multidimensional, Or Configurational?". *Journal of Management & Governance,* 11 (2)**,** 151-163.

Markus, M. L. and Bui, Q. 2012. "Going Concerns: The Governance of Interorganizational Coordination Hubs". *Journal of Management Information Systems,* 28 (4)**,** 163-198.

Markus, M. L., Manville, B. and Agres, E. C. 2000. "What Makes a Virtual Organization Work?". *Sloan Management Review,* 42 (1)**,** 13-26.

Martinez-Torres, M. R. and Diaz-Fernandez, M. C. 2014. "Current Issues and Research Trends on Open-Source Software Communities". *Technology Analysis & Strategic Management,* 26 (1)**,** 55-68.

Masson, P. 2011. "Open Governance in Higher Education: Extending the Past to the Future". *EDUCAUSE Review,* 46 (1)**,** 44.

Mayer, K. J. and Salomon, R. M. 2006. "Capabilities, Contractual Hazards, and Governance: Integrating Resource-Based and Transaction Cost Perspectives". *Academy of Management Journal,* 49 (5)**,** 942-959.

Mockus, A., Fielding, R. T. and Herbsleb, J. D. 2002. "Two case studies of open source software development: Apache and Mozilla". *ACM Transactions on Software Engineering and Methodology (TOSEM),* 11 (3)**,** 309-346.

Morgan, L., Feller, J. and Finnegan, P. 2013. "Exploring Value Networks: Theorising the Creation and Capture of Value with Open Source Software". *European Journal of Information Systems,* 22 (5)**,** 569-588.

Nakakoji, K., Yamamoto, Y., Nishinaka, Y., Kishida, K. and Ye, Y. Evolution Patterns of Open-Source Software Systems and Communities. Proceedings of the international workshop on Principles of software evolution, 2002. ACM, 76-85.

Nicolini, D. 2013. *Practice Theory, Work, and Organization: An Introduction*, Oxford University Press.

Nicolini, D. and Monteiro, P. 2016. "The practice approach: for a praxeology of organisational and management studies". *The SAGE Handbook of Process Organization Studies***,** 110.

Nyman, L. and Lindman, J. 2013. "Code Forking, Governance, and Sustainability in Open Source Software". *Technology Innovation Management Review,* 3 (1)**,** 7.

O'Mahony, S. 2005. Nonprofit Foundations and their Role in Community-Firm Software Collaboration. *In:* Feller, J., Fitzgerald, B., Hissan, S. A. & Lakhani, K. R. (eds.) *Perspectives On Free and Open Source Software.* Cambridge: The MIT Press.

O'Mahony, S. 2007. "The Governance of Open Source Initiatives: What Does it Mean to be Community Managed?". *Journal of Management & Governance,* 11 (2)**,** 139-150.

O'Mahony, S. and Bechky, B. A. 2008. "Boundary Organizations: Enabling Collaboration Among Unexpected Allies". *Administrative Science Quarterly,* 53 (3)**,** 422-459.

O'Mahony, S. and Ferraro, F. 2007. "The Emergence of Governance in an Open Source Community". *Academy of Management Journal,* 50 (5)**,** 1079-1106.

Ojha, A. K. and Rao, R. A. 2014. "The Emergence of an Organizational Field: The Case of Open Source Software". *The Journal for Decision Makers,* 39 (2)**,** 127-143.

Open Source Initiative. 2014. *Open Source Initiative* [Online]. Available: http://opensource.org/ [Accessed].

Orlikowski, W. J. 1993. "CASE Tools as Organizational Change: Investigating Incremental and Radical Changes in Systems Development". *MIS Quarterly***,** 309-340.

Orlikowski, W. J. 2007. "Sociomaterial Practices: Exploring Technology at Work". *Organization Studies,* 28 (9)**,** 1435-1448.

Orlikowski, W. J. and Scott, S. V. 2008. "Sociomateriality: Challenging the Separation of Technology, Work and Organization". *The Academy of Management Annals,* 2 (1)**,** 433-474.

Pentland, B. T. and Singh, H. 2012. Materiality: What are the Consequences. *In:* Leonardi, P. M., Nardi, B. A. & Kallinikos, J. (eds.) *Materiality and Organizing: Social Interaction in a Technological World.*

Poppo, L. and Zenger, T. 1998. "Testing Altermative Theories of the Firm: Transaction Cost, Knowledge-Based, and Measurement Explanations for Make-or-Buy Decisions in Information Services". *Strategic Management Journal***,** 853-877.

Powell, W. W. 1990. "Neither market nor hierarchy: Network forms of organization". *Research in Organizational Behavior,* 12**,** 295-336.

Provan, K. G. and Kenis, P. 2008. "Modes of Network Governance: Structure, Management, and Effectiveness". *Journal of Public Administration Research and Theory,* 18 (2)**,** 229-252.

Puranam, P., Alexy, O. and Reitzig, M. 2014. "What's "New" about New Forms of Organizing?". *Academy of Management Review,* 39 (2)**,** 162-180.

Raymond, E. S. 1998. "Homesteading the Noosphere". *First Monday,* 3 (10).

Raymond, E. S. 1999. "The Cathedral and the Bazaar". *Knowledge, Technology & Policy,* 12 (3)**,** 23-49.

Ren, Y., Kraut, R. and Kiesler, S. 2007. "Applying Common Identity and Bond Theory to Design of Online Communities". *Organization Studies,* 28 (3)**,** 377-408.

Rolandsson, B., Bergquist, M. and Ljungberg, J. 2011. "Open Source In The Firm: Opening Up Professional Practices Of Software Development". *Research Policy,* 40 (4)**,** 576-587.

Sack, W., Détienne, F., Ducheneaut, N., Burkhardt, J. M., Mahendran, D. and Barcellini, F. 2006. "A Methodological Framework for Socio-Cognitive Analyses of Collaborative Design of Open Source Software". *Computer Supported Cooperative Work,* 15 (2-3)**,** 229-250.

Santos, F. M. and Eisenhardt, K. M. 2005. "Organizational Boundaries and Theories of Organization". *Organization Science,* 16 (5)**,** 491-508.

Schaarschmidt, M., Walsh, G. and von Kortzfleisch, H. F. 2015. "How Do Firms Influence Open Source Software Communities? A framework and Empirical Analysis of Different Governance Modes". *Information and Organization,* 25 (2)**,** 99-114.

Seidel, S. and Urquhart, C. 2013. "On Emergence and Forcing in Information Systems Grounded Theory Studies: The Case of Strauss and Corbin". *Journal of Information Technology,* 28 (3)**,** 237-260.

Shah, S. K. 2006. "Motivation, Governance, and the Viability of Hybrid Forms in Open Source Software Development". *Management Science,* 52 (7)**,** 1000-1014.

Shaikh, M. 2016. "Negotiating Open Source Software Adoption in the UK Public Sector". *Government Information Quarterly,* 33**,** 115–132.

Shaikh, M. and Henfridsson, O. 2017. "Governing Open Source Software through Coordination Processes". *Information and Organization,* 27 (2)**,** 116-135.

Shaikh, M. and Vaast, E. 2016. "Folding and Unfolding: Balancing Openness and Transparency in Open Source Communities". *Information Systems Research,* 27 (4)**,** 813-833.

Sharma, S., Sugumaran, V. and Rajagopalan, B. 2002. "A Framework for Creating Hybrid-Open Source Software Communities". *Information Systems Journal,* 12 (1)**,** 7-25.

Singh, P. V. and Tan, Y. 2010. "Developer Heterogeneity and Formation of Communication Networks in Open Source Software Projects". *Journal of Management Information Systems,* 27 (3)**,** 179-210.

Star, S. L. 2010. "This is not a Boundary Object: Reflections on the Origin of a Concept". *Science, Technology, & Human Values,* 35 (5)**,** 601-617.

Stewart, K. J., Ammeter, A. P. and Maruping, L. M. 2006. "Impacts of License Choice and Organizational Sponsorship on User Interest and Development Activity in Open Source Software Projects". *Information Systems Research,* 17 (2)**,** 126-144.

Stewart, K. J. and Gosain, S. 2006. "The impact of ideology on effectiveness in open source software development teams". *MIS Quarterly,* 30 (2)**,** 291-314.

Suddaby, R. 2006. "From the Editors: What Grounded Theory is Not". *Academy of Management Journal,* 49 (4)**,** 633-642.

Teigland, R., Di Gangi, P. M., Flåten, B., Giovacchini, E. and Pastorino, N. 2014. "Balancing on a Tightrope: Managing the Boundaries of a Firm-Sponsored OSS Community and its Impact on Innovation and Absorptive Capacity". *Information and Organization,* 24 (1)**,** 25-47.

Tilson, D., Lyytinen, K. and Sørensen, C. 2010. "Research Commentary—Digital infrastructures: The Missing IS Research Agenda". *Information Systems Research,* 21 (4)**,** 748-759.

Tiwana, A., Konsynski, B. and Bush, A. A. 2010. "Research Commentary—Platform Evolution: Coevolution of Platform Architecture, Governance, and Environmental Dynamics". *Information Systems Research,* 21 (4)**,** 675-687.

Tiwana, A., Konsynski, B. and Venkatraman, N. 2013. "Special Issue: Information Technology and Organizational Governance: The IT Governance Cube". *Journal of Management Information Systems,* 30 (3)**,** 7-12.

Urquhart, C. 1997. Exploring Analyst-Client Communication: Using Grounded Theory Techniques to Investigate Interaction in Informal Requirements Gathering. *In:* Lee, A. S., Liebenau, J. & Degross, J. I. (eds.) *Information Systems and Qualitative Research.* London: Chapman & Hall.

Urquhart, C. and Fernandez, W. 2013. "Using Grounded Theory Method in Information Systems: The Researcher as Blank Slate and other Myths". *Journal of Information Technology,* 28 (3)**,** 224-236.

Urquhart, C., Lehmann, H. and Myers, M. D. 2010. "Putting the 'Theory' Back into Grounded Theory: Guidelines for Grounded Theory Studies in Information Systems". *Information Systems Journal,* 20 (4)**,** 357-381.

Valorinta, M. 2011. "IT Alignment and the Boundaries of the IT Function". *Journal of Information Technology,* 26 (1)**,** 46-59.

Van Grembergen, W., De Haes, S. and Guldentops, E. 2004. Structures, Processes and Relational Mechanisms for IT Governance. *In:* Van Grembergen, W. (ed.) *Strategies for Information Technology Governance.* Hershey PA: Idea Group Publishing.

von Hippel, E. and von Krogh, G. 2003. "Open Source Software and the Private-Collective Innovation Model: Issues for Organization Science". *Organization Science,* 14 (2)**,** 209-223.

von Krogh, G., Haefliger, S., Spaeth, S. and Wallin, M. W. 2012. "Carrots and Rainbows: Motivation and Social Practice in Open Source Software Development". *MIS Quarterly,* 36 (2)**,** 649-676.

von Krogh, G. and Spaeth, S. 2007. "The Open Source Software Phenomenon: Characteristics that Promote Research". *The Journal of Strategic Information Systems,* 16 (3)**,** 236-253.

Walsh, I., Holton, J. A., Bailyn, L., Fernandez, W., Levina, N. and Glaser, B. G. 2015. "What Grounded Theory Is… A Critically Reflective Conversation Among Scholars". *Organizational Research Methods,* 18 (4)**,** 581-599.

Watson-Manheim, M. B., Chudoba, K. M. and Crowston, K. 2012. "Perceived Discontinuities and Constructed Continuities in Virtual Work". *Information Systems Journal,* 22 (1)**,** 29-52.

Watson, R. T., Boudreau, M., Greiner, M., Wynn, D., York, P. and Gul, R. 2005. "Governance and Global Communities". *Journal of International Management,* 11 (2)**,** 125-142.

Weill, P. 2004. "Don't Just Lead, Govern: How Top-Performing Firms Govern IT". *MIS Quarterly Executive,* 3 (1)**,** 1-17.

West, J. 2003. "How open is open enough? Melding proprietary and open source platform strategies". *Research policy,* 32 (7)**,** 1259-1285.

West, J. and O'Mahony, S. 2005. Contrasting Community Building in Sponsored and Community Founded Open Source Projects. Proceedings of the 38th Annual Hawaii International Conference on System Sciences (HICSS'05), 2005 Hawaii. IEEE.

West, J. and O'Mahony, S. 2008. "The Role of Participation Architecture in Growing Sponsored Open Source Communities". *Industry and Innovation,* 15 (2)**,** 145-168.

Wheeler, B. 2003. "Aligning IT strategy to open source, partnering, and Web services". *EDUCAUSE Center for Analysis and Research (ECAR),* 2003 (24).

Wheeler, B. 2004. The Inevitable Unbundling of Software and Support. Available: http://www.immagic.com/eLibrary/ARCHIVES/GENERAL/CMPSTECH/S040301W.pdf.

Wheeler, B. 2007. "Open Source 2010: Reflections on 2007". *Educause Review,* 42 (1)**,** 49.

Whittington, R. 2006. "Completing the Practice Turn in Strategy Research". *Organization Studies,* 27 (5)**,** 613-634.

Winter, S., Berente, N., Howison, J. and Butler, B. 2014. "Beyond the Organizational 'Container': Conceptualizing 21st Century Sociotechnical Work". *Information and Organization,* 24 (4)**,** 250-269.

Xu, J., Gao, Y., Christley, S. and Madey, G. 2005. A topological analysis of the open souce software development community. Proceedings of the 38th Annual Hawaii International Conference on System Sciences (HICSS'05), 2005. IEEE.

Yin, R. K. 2014. *Case study research: Design and methods*, Sage publications.

Yoo, Y. 2013. "The Tables have Turned: How can the Information Systems Field contribute to Technology and Innovation Management Research?". *Journal of the Association for Information Systems,* 14 (5)**,** 227.

Appendix A: Sample Interview Guides

Interview Guide-Admin2

1. Could you please describe your roles and responsibilities in Kuali community?
2. "Keeping your money in your mission" is one of the main purposes for creating Kuali community. From my understanding, it means aligning Kuali with the strategic plan of universities. Could you please explain how is Kuali aligned with higher education?
3. Can you please elaborate on the process of membership?
4. What is the difference between a member and a partner?
5. Do you keep track on Kuali adopters; those who download Kuali projects without contributing back or joining the community?
6. In Kuali documents, another group of the community are called 'lone wolves'. Who are they? How do you identify them?
7. What are the roles of the constituency members?
8. So now we have the Kuali products, each is considered as a separate project. Kuali foundation is governing the overall thing. So are the membership financial contribution used for Kuali projects as needed or they are directed on projects?
9. Are Kuali projects governed in the same way, or each project has its own rules and regulations?
10. Does Kuali have an assessment tool for its governance? Does it follow a certain governance framework?
11. Attracting universities to join the community is mostly done from cost perspective. Is it just the cost that distinguishes Kuali from other software development solutions?
12. How Kuali policies are made? Did the policy making mechanisms changed as Kuali community grew? (The formality of policy making).
13. Kuali website contains mainly content-related policies. Are there any behaviour-related policies?
14. Monitoring is an essential concept in governance. How do you monitor the behaviour of individuals, universities and KCA to check whether they are adhering to Kuali values and practices?
15. The peer-reviewing nature of HE, the collaboration and capacity building are all characteristics of HE, and in the same time are main values of OSS community. From you point of view, is the nature of HE played any role in governing Kuali?

16. As you know, opening up the code for the public is one way for inviting innovation to the software development process. Kuali adds sort of restrictions on code contributions by setting rules and requirements for the contributed code. This is also obvious in the absence of volunteers from outside the community, and not accepting code contributions from outsiders (or their contribution is placed under scrutiny). How does that impact on the innovation and creativity?

17. What are the mechanisms applied to avoid Kuali projects from being dominated by the requirements of certain universities?

18. Do Kuali face situations where one or more member has specific requirements that are not fulfilled by Kuali? Do they implement the changes on their own without contributing back? How does Kuali handle this?

19. Commercial affiliates played an essential role in building and sustaining Kuali community. We know that CA have their own agenda that contradicts with Kuali values in terms of maximizing profits. How does Kuali fulfil these contradictory agenda and maintain Kuali values at the same time?

20. In terms of control, OSS solutions allows universities to have control over their software in all aspects. In the same time Kuali foundation has imposed some sort of control over the community to ensure the quality of the product. So, how does Kuali balance between empowerment and control?

21. I've noticed that Kuali projects are less likely to be forked. Why is it the case?

22. With regards to the public documents of Kuali website, wiki, KIS, are they accurate and reflect Kuali community? How frequent are they updated?

23. Kuali turned into 10 years old in August 2014, and it was considered as a turning point in Kuali's life. Can you please briefly talk through how did you get back to the community to build a new Kuali strategy?

24. Did any universities leave Kuali? Why?

25. Can you please guide me on who else can I interview to know more about the community?

26. Is there anything else you think is useful for my research on governance and I did not cover in the interview?

27. Is there anything you would like to ask me?

Interview Guide-Admin3

1. Let's start with the event that happened in 2005/2006 when MIT decides to give the intellectual property of MIT Coeus to Kuali. What was MIT's motive behind that decision?

2. The structure of Kuali was initially built to accommodate KFS. When Coeus joined the community, did it had to inherit the existing structure and development framework, or you had to build your own?

3. You have moved from having a local development team to an ecosystem of pooled resources. What are the actions taken to cope with overhead tasks and time commitment?

4. Are there any KC partners outside the US? If yes, what is the experience? If no, why not?

5. As a member of Kuali Boards, what are the changes that have occurred in Kuali priorities?

6. So, now what are the plans with regards to KualiCo, how would this change

7. Is there anything else you think is useful for my research on governance and I did not cover in the interview?

8. Is there anything you would like to ask me?

Interview Guide-Tech2

1. Can you please describe your roles and responsibilities in Kuali community?

2. From my observation on Kuali community, I have noticed that there are 2 main tools used to organise the developers work: Jira, and Fisheye. Can you please summaries how each work in terms of your day-to-day tasks. Let's start with Jira.

3. What is the role of watchers?

4. So, the communication in Jira is between developers within the same universities, or do you have communication between developers from different uni's?

5. Can you summarise the role of fisheye?

6. Up to what extent the developers (technical teams) have a say on Kuali direction?

7. Are there any monitoring tools to ensure that the code is written according to standards?

8. Is there a high reliance on core developers? Limited number of developers performing large amount of tasks?

9. With regards to KualiCo, what are your current responsibilities towards Kuali?

10. Is there anything else you think is useful for my research on governance and I did not cover in the interview?

11. Are there any other Kuali members that you think that may be beneficial for my research?

12. Is there anything you would like to ask me?

Appendix B: Meeting Summaries

Meeting code	Date
Meeting1	Jan-10
Meeting2	Feb-10
Meeting3	Mar-10
Meeting4	Apr-10
Meeting5	May-10
Meeting6	Jun-10
Meeting7	Jul-10
Meeting8	Sep-10
Meeting9	Oct-10
Meeting10	Nov-10
Meeting11	Dec-10
Meeting12	Jan-11
Meeting13	Mar-11
Meeting14	Apr-11
Meeting15	Jun-11
Meeting16	Jul-11
Meeting17	Sep-11
Meeting18	Oct-11
Meeting19	Nov-11
Meeting20	Jan-12
Meeting21	Apr-12
Meeting22	May-12
Meeting23	Jul-12
Meeting24	Aug-12
Meeting25	Sep-12
Meeting26	Oct-12
Meeting27	Dec-12
Meeting28	Jan-13

Meeting code	Date
Meeting29	Feb-13
Meeting30	Mar-13
Meeting31	Apr-13
Meeting32	May-13
Meeting33	Jul-13
Meeting34	Aug-13
Meeting35	Sep-13
Meeting36	Oct-13
Meeting38	Dec-13
Meeting39	Summer 2013
Meeting40	Jan-14
Meeting41	Feb-14
Meeting42	Apr-14
Meeting43	Jun-14
Meeting44	Jul-14
Meeting45	Aug-14
Meeting46	Oct-14
Meeting47	Dec-14
Meeting48	Jan-15
Meeting49	Feb-15
Meeting50	Apr-15
Meeting51	May-15
Meeting52	Sep-15

Sample Meeting Summaries

Kuali Foundation Board Phone Call
01-08-2010, 2:00pm ET. 11:00am PT

Attending:

[List of attendees]

Not attending:

[List of absentees]

1. **FOR VOTE: Review and approve minutes from 12/11/09 meeting**
[redacted] moved, [redacted] seconded, and the minutes were approved by all.

2. **FOR VOTE: Validate election of Board Officers**
We validated our recent Officer elections, with [redacted] a Chair, [redacted] as Vice Chair, [redacted] as Secretary, and [redacted] as Treasurer. [redacted] moved, [redacted] seconded, and the new Officers were approved by all.

3. **FOR ACTION: Update language in minutes on KC Sustainment Model**
We discussed the recent e-mail about needing to change the timing of the 2-year review from the implementation of the KCC to the convergence itself. Everyone agreed.

4. **FYI: Role of Foundation Board Members**
[redacted] indicated that he has sent in the mail to each Board member a couple of small books/pamphlets about role of not-for-profit boards. It might be helpful to discuss that at a future meeting.

5. **FYI: Resolution on Rice Funding Model**
We had voted on this and approved it. The Rice Board will report back the progress on the model to this Board.

6. **FOR DISCUSSION AND ACTION: Survey of potential members from Kuali Days**
We don't want to be perceived as a hard sell. We could do a survey for those we don't know well. We could contact directly those we already have a relationship with. We agreed that not too much effort should be involved and the survey should be short and easy. We questioned if the timing is right vis-à-vis the Mellon RIFT issue? Maybe we wait to do the survey for 30-45 days. [redacted] will handle this from here, by parsing the spreadsheet to ensure follow-up conversations are happening and delivery of a survey for those we don't have direct contact with.

7. **FOR DISCUSSION AND ACTION: Travel "Designated Dues" (attached)**
The advice from the Board was that we want to be fully above board on transparency and perception, and not be seen to circumvent policies. Can we take a step back and look at the problem of needing to have F2F meetings in the future, that schools may not be able to fund directly.

8. **FYI: Principles for Incubated and Approved Projects**
[redacted] reiterated our current practice for the record and for new Board members:
   a. Incubated Projects must have a Kuali Foundation Board advocate.
   b. In order to be approved, Kuali Projects must create a Charter, working with the Executive Director, and that Charter must be approved by the Kuali Foundation Board.

It was noted that for some new projects that are considered modules of existing projects (one example is Materials Management), these would not go through incubation and approval, but be under the project Board's aegis. Sometimes this may be a gray area, so we'll need to be flexible and look at each case-by-case.

9. **FYI: Items pending -** [redacted] will follow up aggressively:
--Policies update
--Pricing Committee
--Open a thread about having a conference committee for Kuali Days IX
--Open a thread about Board Transparency
--Working on Treasurer transition with [redacted] and [redacted]

10. **Open discussion**
[redacted] indicated that the University of Illinois is interested in becoming a member and they have questioned whether Urbana's membership has an effect on other campuses. We all agreed that we would continue the practice that allows either a system membership or a campus membership. Either way, the membership comes with one seat and one contact to Kuali.
[redacted] expressed the sincere hope that all Board Members can attend the F2F in May at CSU. We only have two F2F meetings of the Board per year, so it is critical that we all attend.
[redacted] asked if people would think about pursuing BI separately by project or as a whole. There was some discussion about the scope of this potential project, and [redacted] asked anyone interested in talking further to contact him.
[redacted] indicated that at our next meeting, the Kuali Team will be joining us to talk about a new system for managing documents, people, tools, access, etc.

11. **Next meeting is February 8, 2010**

Kuali Foundation Meeting at Kuali Days
2011 JW Marriott Hotel, Indianapolis
Thursday, November 17, 2011, Noon-6pm ET

Attending:

[List of attendees]

Not Attending:

[List of absentees]

1. Review and Debrief on Kuali Days 2011

Input on Program and Sessions:

--we'll need to review evaluations to see if we have too many sessions and if all are high quality
--we need to clearly separate sessions that teach and demo software versus implementation stories
--combining two sessions into one did not work out well
--Track Chairs should manage very carefully the overall curriculum and fill either by CFP **or** assignment

Length and Logistics:

--move to one keynote speaker
--we decided we could go to 2.5 days (probably in 2013, as 2012 is already set)
in future --we won't silo projects on particular days because cross pollination is important
-continue to have Board meetings outside of sessions to allow for networking during Kuali Days

Overall:

-we don't have good enough support for tire kickers and newcomers
-we might consider: Chair/Vice Chair panels, "come and meet XXX", speed-dating, CFO and CIO roundtables, a Pre-Conf to sit down with a Board, lunch tables for SIGs, etc.
-collaboration showcases are good
-consider tweets running on monitors
-market KD at other conferences and ensure the message is that we're not a technical conference
-have amateur videography, small video captures, maybe even a booth to do this.

2. Kuali FY2010 Financials

3. Review and Approve October Meeting Summary

4. Review of the Foundation Fund financials

5. Licensing Audit

The report for 2010 was shared. All issues were resolved. At this time, we need to consider if we start the process for 2012, since our goal is to do this every two years.
We need to be stronger in this area. There have been personnel transitions in this area and therefore, Kuali lacks the intensity of focus, expertise, and education it needs. [redacted] and [redacted] will be discussing this and report back to this Board with recommendations.

ACTION ITEMS:

--[redacted] and [redacted] will convene to discuss and report back to this Board.
-[redacted] will correct the title on the Audit report and file in our Documents of Record.

6. Growing the KCA Community

The group discussed potential additional partnerships and exploration of additional KCAs. Brad will follow up in this area and report to the Board of any developments.
We agreed that we should continue to support current KCAs so they are healthy. To grow, we should look for new KCAs who can fill holes that the community would value.

7. Recruiting Smaller Schools

Key issues:

--Kuali needs the full suite of Financials, HR/Payroll, and Student because small schools usually cannot go best of breed. We need to show a truthful roadmap for KPME and KS.
--Kuali needs to have offerings in the cloud.
--Kuali needs to market to small schools because much of this is myth busting. We have San Joaquin Delta, Naval Postgraduate School, Stevens Institute, Haverford College, and others to demonstrate Kuali is viable for small schools.
-[redacted] shared what KFS is doing to address this —to create a supplemental partner model to get more diversity in representation. Other projects could use that model.

ACTION ITEMS:

--[redacted] and [redacted] will start a process for outreach here, using NPS, Stevens, SJDC and Haverford as examples.

Appendix C: Sample Emails

Technical Emails

Sender: [redacted] **Date:** 17/12/2013
Good morning all,
Reaching out because I'm trying to troubleshoot a pesky issue we are facing in KFS 4.1.1 and I haven't been able to find anything in the JIRA specifically about it. We have intermittent Doc Search failures in our production environment periodically. They happen on random nodes (env is load balanced) and it seems to happen most often in the mornings when users begin signing into the KFS system and doing their morning doc searches. This is not happening in any of our other environments.
Here is the odd part. A restart of the offending Tomcat node "fixes" the issue. Once we restart Tomcat, the issue will be resolved for the rest of the day/week until it randomly occurs again.
Has anyone seen this issue or something similar to it? I'm investigating a root cause since the Tomcat restarts are only an interim fix that we're doing as a reaction to the error happening. I've placed some of the stack traces in-line below.
Any help would be greatly appreciated. Thanks!

Reply1: [redacted] **Date:** 17/12/2013
IIRC, I've also seen this problem a few months ago, but haven't seen it in a while. We're still developing KFS, so we're not seeing the heavy load that you are. We're using solaris hardware. Our tomcats are not (yet) load-balanced. Perhaps a java/tomcat upgrade fixed the problem for us. We're currently using java 1.7.0_45 and tomcat 6.0.36.

Reply2: [redacted] **Date:** 02/03/2014
If I remember correctly, you're running KFS with Rice embedded and load-balanced. Is this log information coming from the KFS instance or the Rice instance?

Reply3: [redacted] **Date:** 06/03/2014
OK, I *think* that I've found the problem.
DocSearchCriteriaDTOLookupableHelperServiceImpl has the following line in it:
static DatabasePlatform dbPlatform = (DatabasePlatform) GlobalResourceLoader.getService(RiceConstants.DB_PLATFORM); When the KFS server started, that line threw an NPE. Since the static initializer threw an exception, the classloader failed to load that class, and that problem persisted. My suggestion is to make dbPlatform an instance variable and set the value in the constructor instead.

Sender: [redacted] **Date:** 07/01/2014
Friends, We are discussing upgrades to our MySQL 5.5.19 KFS and Rice databases, and are wondering if there are institutions currently running KFS and Rice on 5.6 or newer. Could you please let me know if your institution is?

Reply1: [redacted] **Date:** 10/01/2014
Hi Doug. I wanted to point out to you https://jira.kuali.org/browse/KFSMI-11610, which is a contribution Cornell has made to KFS 5.1.2; this is a copy of a fix for impex that will allow the impex process to run with MySQL 5.6. (ie, KFS doesn't currently support MySQL 5.6 but theoretically will soon). The good news here, though, is that you're not building a new db from scratch; you're updating a db. The other good news is that it's highly likely that some developers for Rice or other client apps have been running MySQL 5.6 since Rice fixed this issue a few months ago if memory serves. The bad news, on the other hand, is that, due to the silence in response to your question, it seems unlikely that anyone is running KFS against MySQL 5.6 today. We at foundation certainly aren't. Hope this helps somewhat, though.

Reply2: [redacted] **Date:** 11/01/2014
Thank you, James.

Reply3: [redacted] **Date:** 11/01/2014
I'd like to point out KFSMI-11610 also effects MySQL 5.5. Since Haverford is upgrading from 5.5, they probably won't need this fix. Here is the reference info from the Mysql 5.5 manual http://dev.mysql.com/doc/refman/5.5/en/innodb-restrictions.html Also, here is the effected Rice issue. https://jira.kuali.org/browse/KULRICE-9021

Reply4: [redacted] **Date:** 11/01/2014
FWIW I can say that running KFS with MySQL 5.6 locally (on my computer) seems to be functioning well for me (and I think for others here at IU, too).

Sender: [redacted] **Date:** 01/12/2014
Good morning all,
We are attempting to run KFS 5.3 with standalone RICE (2.3.6). I have been searching the upgrade documentation and config documentation on the Wiki. Is there one particular document anywhere that shows the configuration and steps needed to setup a ""standalone"" environment (KFS and RICE on separate nodes, running modes as EMBEDDED and REMOTE) versus the bundled mode? I only see a few references to ""do X if running standalone"" rather than an authoritative setup doc.
If anyone knows the location of such a thing, a link would be much appreciated.

Reply1: [redacted] **Date:** 01/12/2014
Ben,
There is this page on Standalone Rice: https://wiki.kuali.org/x/BQACEw
There are only a few properties listed, because if you set those, particularly rice.standalone to true and rice.url to your base URL for the Rice standalone server, and rebuild KFS (ant dist or dist-local) the configuration will be setup to support standalone Rice as part of the build process. (see also the Setup 5: https://wiki.kuali.org/x/lAACEw and Build 5: https://wiki.kuali.org/x/bgACEw pages in the KFS Documentation Confluence space).
Hope this helps,

Reply2: [redacted] **Date:** 01/12/2014
Thanks for the quick response [redacted]. This is very helpful info!

Functional Email

Sender: [redacted] **Date:** 17/01/13

Hello, Is there a way to run all of the nightly batch jobs on demand? I don't want to run them one by one via the unscheduled jobs, but want to essentially initiate the entire nightly job process at once, if possible. Is there any way to do that? We would be doing this in one of our test environments, not production.

Reply1: [redacted] **Date:** 17/01/13

[redacted], Just manually launch the unscheduled scheduleJob. That should kick off all the subordinate nightly jobs.

Reply2: [redacted] **Date:** 17/01/13

Excellent! Thank you!

Reply3: [redacted] **Date:** 17/01/13

[redacted], Please note the potential consequences of such an action:

* ALL the nightly jobs would run - not just a subset

* The system should essentially be pseudo-offline while that's happening (you don't want to have data being entered while the jobs are running)

* If something goes amiss, you wouldn't be able to stop the next jobs from running

It's something I've done a number of times - usually on weekends or after end of daily use (while I'm testing and before the regular nightly jobs kick off). So, it takes thought - and maybe preparation - but yes, it can be done.

Reply4: [redacted] **Date:** 17/01/13

Gotcha. Thank you.

Sender: [redacted] **Date:** 24/11/14

Good Evening, I was wondering if any of the schools made a customized KFS portal or main menu for end users. It could be either within or outside of KFS. If so, would you be willing to send me a screenshot or any other information you can share (experiences etc.)? I'm compiling some ideas together and would like to see what others have done.

Reply1: [redacted] **Date:** 25/11/14

[redacted], I've seen both a customized main menu (done for Stevens) and a menu system that is dynamically driven after logon by user roles. I can't recall the school with the later, but I'm sure someone else will share that. I believe the code and logic was even contributed in some form. Cheers!

Reply2: [redacted] **Date:** 25/11/14

[redacted], We actually have a little of both at Haverford -- menu options driven by roles and some additional minor customizations. Happy to share code if you like. What we don't have, but would like, is a true user friendly portal (which I'm guessing is what you are really after).

Reply3: [redacted] **Date:** 25/11/14

[redacted], I've attached a screenshot showing part of our portal and the menu we've added to Ekualiti KC (our cloud-hosted version of Kuali Coeus) to make navigation more accessible and intuitive. Even though the screenshot shows KC, everything you see applies to KFS just the same. Obviously there is a total UI overhaul. The menu itself has the following features:

- Pre-loaded => FAST!
- Hierarchical; saves space by eliminating many text links
- Available on all screen (unlike KFS's portal page)
- Role-driven: only shows options for which user has permission(s)
- Direct links to "Create New" eDocs; that what the "+" are for

(Snapshot)

I think the screenshot shows what can be done if you're willing to depart from the out-of-the-box UI. Let me know if you have any questions. Thanks,