Spring 3-19-2020

Influence Of Developer Sentiment And Stack Overflow Developers On Open Source Project Success: An Empirical Examination

Johnson Rajakumar

PERMISSION TO BORROW

In presenting this dissertation as a partial fulfillment of the requirements for an advanced degree from Georgia State University, I agree that the Library of the University shall make it available for inspection and circulation in accordance with its regulations governing materials of this type. I agree that permission to quote from, copy from, or publish this dissertation may be granted by the author or, in her absence, the professor under whose direction it was written or, in his absence, by the Dean of the Robinson College of Business. Such quoting, copying, or publishing must be solely for scholarly purposes and must not involve potential financial gain. It is understood that any copying from or publication of this dissertation that involves potential gain will not be allowed without written permission of the author.

Johnson Rajakumar

NOTICE TO BORROWERS

All dissertations deposited in the Georgia State University Library must be used only in accordance with the stipulations prescribed by the author in the preceding statement.

The author of this dissertation is:
Johnson Rajakumar
4295 Noor View Court
Johns Creek
Georgia GA 30022

The director of this dissertation is:

Yusen Xia
J. Mack Robinson College of Business
Georgia State University
Atlanta, GA 30302-4015

Influence of developer sentiment and Stack Overflow developers on Open Source Project Success: An Empirical Examination

by

Johnson Rajakumar

A Dissertation Submitted in Partial Fulfillment of the Requirements for the Degree

Of

Doctorate in Business Administration

In the J. Mack Robinson College of Business

Of

Georgia State University

GEORGIA STATE UNIVERSITY
J. MACK ROBINSON COLLEGE OF BUSINESS
2020

ACCEPTANCE

This dissertation was prepared under the direction of the *JOHNSON RAJAKUMAR* Dissertation Committee. It has been approved and accepted by all members of that committee, and it has been accepted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Business Administration in the J. Mack Robinson College of Business of Georgia State University.

Richard Phillips, Dean

DISSERTATION COMMITTEE

Dr. Yusen Xia Ph.D. (Chair)

Dr. G. Peter Zhang, Ph.D

Dr. Ling Xue, Ph.D

DEDICATION

This work is dedicated to my wife Dian and my four children Jaedon, Jerusha, Jotham, and Johanna, without whom this work could not have been completed. I am grateful for your support and understanding over the last three years. I want my sons and daughters to come back to this dissertation with the knowledge that everything is possible, and if one can work hard to achieve it.

ACKNOWLEDGEMENTS

"Some trust in chariots and some in horses, but we trust in the name of the LORD our God." Psalm 20:7

I thank God for allowing me to pursue the doctorate degree, who has blessed me with the gift of knowledge and understanding, so that I may benefit my family members and others to attempt excellence in everything.

I also express my gratitude to my committee chair, Dr. Yusen Xia, and my committee members, Drs. Peter Zhang and Dr. Ling Xue, for their support throughout this research. I thank Dr.Xia for inspiring me and guiding me during this study.

I thank the program leadership, Dr. Lars Mathiassen, Dr. Louis J. Grabowski and Jorge Vallejos for their continued guidance through this remarkable journey.

I acknowledge my 2020 cohorts for providing their support and sharing their knowledge over the past three years.

I also thank my wife, Dian, for providing spiritual encouragement to commence this educational journey.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

GH	: Github
OSI	: Open Source Integration
OSS	: Open Source Software
SO	: Stack Overflow
IT.	: Information Technology
D&M	: Delone & McLean

ABSTRACT

Influence of developer sentiment and Stack Overflow developers on Open Source Project Success: An Empirical Examination

By

Johnson Rajakumar

April 2020

Chair: Dr.Yusen Xia

Major Academic Unit: Executive Doctorate in Business

The collaborative effort of software developers around the world produces Open Source Software (OSS) products, and most importantly, the source code of the software product is shared publicly. A recent survey of 1300 IT professionals by Black Duck Software showed that the percentage of companies using open source software grew from 42% to 78% between 2010 and 2015 (Anthes, 2016). There has been a significant increase in the formation of self-organizing virtual teams to produce open source software products and services. The current literature does not address the factors affecting the success of open source projects through the lens of self-organizing virtual teams and the sentiment among software developers. This phenomenon suggests a need to understand how successful project teams are created in a virtual collaborative environment.

This research investigates how successful virtual teams are formed through the influence of an online developer community. The focus of this research is to assess how the online developer community, Stack Overflow (SO), influences the success of open source projects. More precisely, the study empirically tests the influence of the SO community on successful Github (GH) projects. The investigation also empirically examines how the ties among the software developers in the SO community initiate the self-creation of OSS project teams. The research also explores the perception of the developers about open source projects. Furthermore, the study probes the impact of OSS artifacts, namely “feature” and “patch” requests, on open source projects.

The findings indicate that the perception of the developers in the SO community, prior ties among the developers in the community, and the artifact type of the project are the factors that influence the success of OSS projects. The research discusses the implications of the outcomes concerning self-organizing open source project teams.

INDEX WORDS: Open Source Projects, Stack Overflow, Virtual Team formation, Developer Sentiment

I INTRODUCTION

I.1 Problem: Formation of successful self-organizing open source project teams

The Open Source Software (OSS) platform enables innovation by sharing skills and ideas from the software developers and application architects. The OSS framework not only promotes collaboration and innovation but also generates significant revenue for the technology industry. The most common business model is the "Dual Licensing Model" in which the software product is distributed not only with the "Open Source Integration" (OSI) license but also with a chargeable commercial product license. The famous OSS projects such as Mongo database and LINUX operating system were successful in the retail market (see Figure 1). Although there are numerous OSS projects in the market, only a few of them have been successful and have produced revenue (Chengalur et al. 2003).

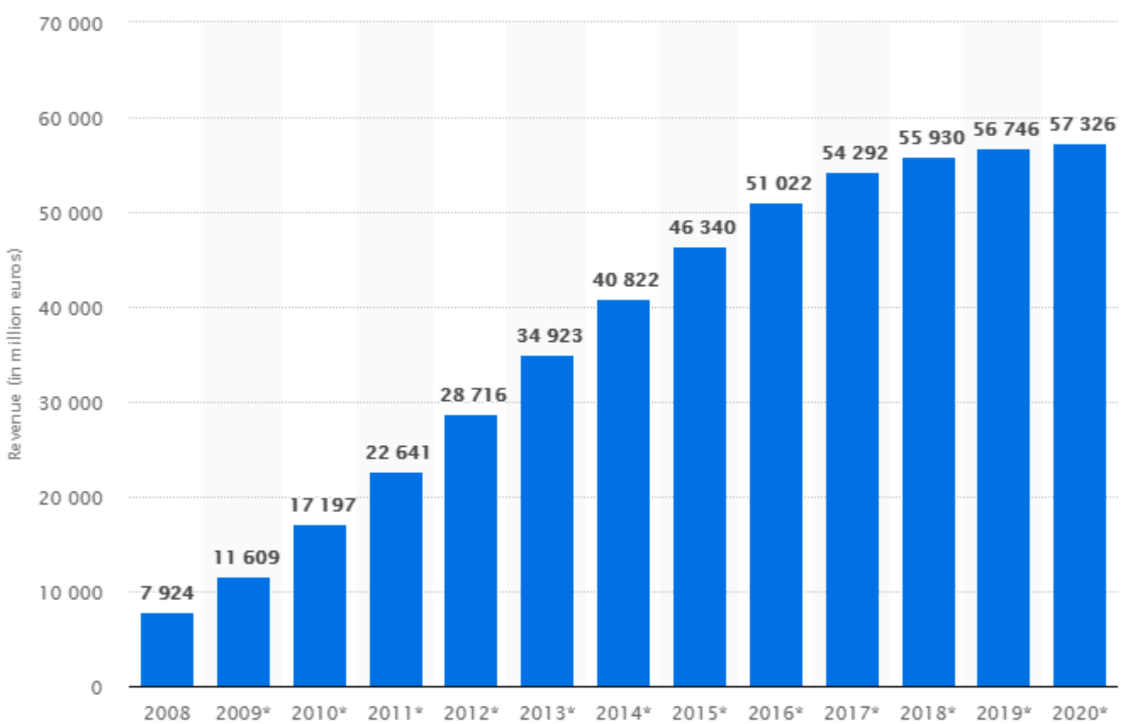


Figure 1 Open Source Market Trend

I.2 Determinants of OSS project success

The success of OSS projects has been ascribed to several OSS characteristics such as operating systems, restrictive licenses, and software type. The knowledge sharing of technical expertise among the project team members is a critical element in the OSS framework. Network social capital has been defined by Portes (1998, p.6) as the “ability of actors to secure benefits by their memberships in social networks or other social structures.” Internal cohesion (cohesion among the project members), external cohesion (cohesion among the external contacts of the project) and technological diversity (resources with diverse technical skillsets) are the significant attributes of open source collaboration networks (Singh et al. 2011). Given the importance of knowledge sharing among the project team members, it is surprising that little research has been performed on network social capital aspects of the project team (an exception being the work by Singh et al. 2011). Besides, OSS research has been centered on using "projects" as the unit of analysis (Rajdeep et al. 2006). The open source projects consist of teams that generate artifacts such as *Feature Requests* (introducing new functions) and *Patch Requests* (bug fixes to existing products) (Temizkan et al. 2015).

I.3 Group formation

The success of OSS projects has prompted many companies to take advantage of the OSS model of development (Stewart et al. 2006). For enterprises, OSS development is a big shift from proprietary software development as the former is characterized by a team of individual developers across different organizations. As such projects evolve, it is essential to understand how the teams are formed and whether they are successful. Team formation is a social phenomenon, and the findings imply that homophily and network constraints based on the existing strong ties exert a strong influence on team composition (Ruef et al. 2003).

I.4 Developer Communities

Community denotes a group of people having a similar set of motives. The information technology (IT) industry has witnessed considerable growth over the last two decades. As complex software solutions require the capturing and sharing of technical knowledge, there is a need for software developers to ask technical questions and receive answers from a community of software engineers. Online software developer forums serve as excellent platforms to share knowledge among the community. The developers use such forums not only to discuss problems but also to share and receive feedback on high-level technical architecture. Stack Overflow (SO) website hosts the software development community, and the platform facilitates the posting and receiving of answers to challenging issues by the developers. The platform offers the right level of quality control by evaluating the posts through feedback from the original poster and by assigning categories.

The advent of social media has dramatically changed the way people express their opinion on the goods and services received from a vendor. As OSS projects evolve, the adaptability of the product depends on the evaluation provided by the software developers. The developers express their opinion through comments and advices in the OSS project hub. Defects or crashes in the software will result in negative reviews by the developers, which will in turn lead to the failure of the product. Developer sentiment plays a pivotal role in the adaptability and success of OSS products.

In this research, the emergence of self-organizing open source project teams from online developer communities has been investigated. Besides, the correlation between successful OSS projects and self-organizing teams from the online developer communities has been explored. This context is significant as it helps us to fathom how the existing relationships in a community affect team formation in the context of a structured project. This context also assists the practitioners in understanding team formation mechanisms that impact the success of OSS projects. Besides, the impact of developer sentiments among the stack overflow community on open source projects has also been studied.

I.5 Purpose of the study

The focus of the research is to examine the effect of stack overflow community and developer sentiment on the success of open source projects.

In this study, the following questions have been addressed:

RQ1: Does the participation of stack overflow community developers influence the success of open source projects?

RQ2: How does the level of participation of stack overflow community developers impact the success of open source projects?

RQ3: Does developer sentiment towards open source projects influence the success of the projects?

RQ4: Do both positive and negative sentiments influence the success of open source projects in the same way?

This study involves artifact-level analysis with multiple programming languages (*C++, Javascript, and Python*) as the *network boundary*. In this work related to OSS, "*Project*" will be used as the unit of analysis. This research contributes to open source industry literature on the behavior of self-organizing teams in a collaborative network and adds to the knowledge of artifact-level analysis.

I.6 Research Structure and Approach

The structure of this research is based upon five elements, namely, P (Problem situation), A (Area of concern), F (Conceptual framing), M (Method), RQ (Research question), and C (Contributions) (Mathiassen et al. 2012). These research elements are described in Table 1.

Table 1: Composition elements of the research study

P (Problem Setting)	The collaborative effort of software developers around the world produces OSS products, and most importantly, the source code of the software product is shared publicly. Open source platforms enable innovation by the sharing of skills and ideas from the software developers and application architects. The knowledge sharing of technical expertise among the project team members is a critical element of the OSS framework. Although the importance of knowledge sharing among the project team members is understood, there is a need to appreciate the importance of self-organizing virtual teams in open source projects. The problem setting for this research is the influence of the online developer community on the success of open source projects.

A (Area of Concern)	The influence of the Stack Overflow community on open source project success
F (Conceptual Framework)	Social Network Theory
M (Research Method)	Quantitative analysis of developer participation from Stack Overflow database and open source project data from Github archive database
RQ (Research Questions)	RQ1: Does the participation of Stack Overflow community developers influence the success of open source projects? RQ2: How does the level of participation of stack overflow community developers impact the success of open source projects? RQ3: Does the developer sentiment towards open source projects impact the success of these projects? RQ4: Do both positive and negative sentiments influence the success of open source projects in the same way?
CP (Contribution to Practice)	• Assessment of the online developer community and the directions for

	future team building through developer communities • Contribution to engaged scholarship on building virtual project teams for enterprises through a pool of talented resources from online developer communities • Development of new recruiting tools and processes to apply within this context • Technical recruitment
CA (Contribution to Area of Concern)	• Detailed empirical research on the influence of developer communities on open source projects • Empirical assessment of developer sentiments participating in open source projects from the developer community. • Contribution to open source industry literature on the behavior of self-organizing teams in a collaborative network

	• Contribution to the area of open source projects and the associated success factors

The current literature lacks an empirical validation of the influence of developer sentiment and stack overflow on open source project success. In this study, datasets collected from the Github and Stackoverflow databases were employed to test the hypotheses. Text mining on user comments was performed in the study to examine the influence of developer sentiments on open source project success.

I.7 Summary

In this section, the structure of the rest of the dissertation has been provided.

Chapter 2: Literature Review

Chapter 2 reviews the theoretical and empirical literature on open source projects and the determinants of project success, with a special focus on the online developer community and developer sentiment. This chapter provides the evidence for the study of OSS determinants of success. This section analyzes the gaps in literature pertaining to the study of group formation and developer sentiment in the context of the online developer community.

Chapter 3: Theoretical Framework

This chapter describes the social network perspective of OSS project development through the lens of network theory. This part also explains the development of hypotheses to evaluate the influence of stack overflow and developer sentiment on OSS projects.

Chapter 4: Research Design and Methodology

This chapter covers the research design, data collection, transformation and analysis, text analysis approach, and methods. This section validates the hypotheses about the research question and provides a detailed description of control, moderator, and dependent and independent variables used in the analysis.

Chapter 5: Results

This chapter furnishes the results of the empirical research and illustrates the output of the descriptive and regression analysis. The results establish the validity of the six hypotheses and provide a successful model. The results of the study successfully validate the relationship between the stack overflow developer community and developer sentiment in open source projects.

Chapter 6: Discussion

This chapter presents the findings and implications of the study. The key findings are analyzed through a theoretical lens. This part also discusses the various contributions of the study to the engaged scholarship and theory. Besides, it lists the limitations of the research and suggests further theories concerning open source project success and online developer communities.

II LITERATURE REVIEW

Researchers, scholars, and corporations have been interested in identifying the determinants of OSS projects as they significantly influence the financial, legal, and policy decisions of the OSS development model (See Table 2). Our review focuses on the literature concerning OSS, starting with information system success determinants, OSS project success measures, OSS project developer network formation, online developer communities and developer sentiment before examining the literature gaps (See Figure 2).

Table 2 Article Summary

	Article 1 (DeLone)	Article 2 (Ravi Sen et al.)	Article 3 (Subramaniam et al.)	Article 4 (Grewal et al.)	Article 5 (Temizkan and Ram L Kumar).	Article 6 (Singh et al.)
OSS Measures of Success	IS success Factors: System Quality, Information Quality, System use, User Satisfaction, Individual Impact, Organizational Impact	Subscriber Base, Developer Base	Relationship among the success factors, Developer Interest in the Project, project activity, user interest	Technical Achievements of a Project as well as indicators of Market or Commercial success	Knowledge Creation - # of CVS Commits	Knowledge Creation - # of CVS Commits
Determinants of OSS Success		Number of Subscribers in a time period Number of Developers in a time period	OSS Licenses (restrictive)	Project age and Number of Page views	Internal Cohesion, External Cohesion, Network Location, Network Decomposition	Internal Cohesion, External Cohesion, Technological Diversity
Variables – Time Invariant		OSS license, Operating System, Programming Language, Accepts financial Donations, User Type	OSS License type, Operating System and Programming language		Programming Language	
Variables – Time Dependent		Project age, Number of developers working on the project in a month (Developers)	Project status, Developer Interest, user interest, and Project Activity		Patch and Feature Request – Repeat Ties, External Cohesion	Repeat Ties, Network Constraint Projects

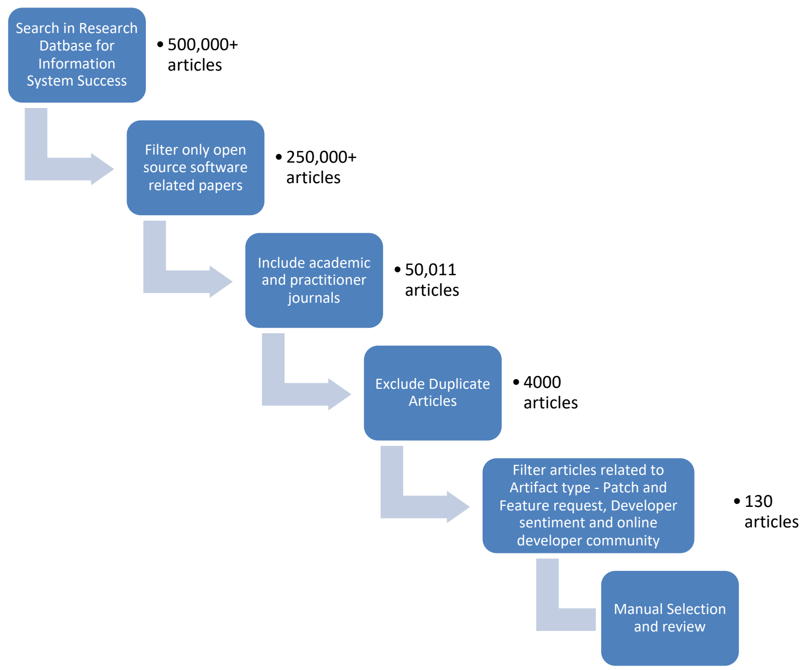


Figure 2 Literature Review Design

II.1 Information systems success determinants

The extensive literature on Information Systems (IS) has focused on various measures to determine the success of an IS project. The most frequently used model for deciding IS success is the one proposed by DeLone and McLean (1992, 2002, 2003). This model provides six interrelated measures of success for IS: *System Quality, Information Quality, System Use, User Satisfaction, Individual Impact, and Organizational Impact*. The model states that the six measures of success are interrelated rather than independent (DeLone et al. 2003). As the role of IS changed over the years, the researchers suggested three major dimensions for IS success: "Information Quality," "Systems Quality," and "Service Quality." They argue that each of the three dimensions should be measured and controlled separately. The Delone and McLean (D&M) IS success model is described in Figure 3.

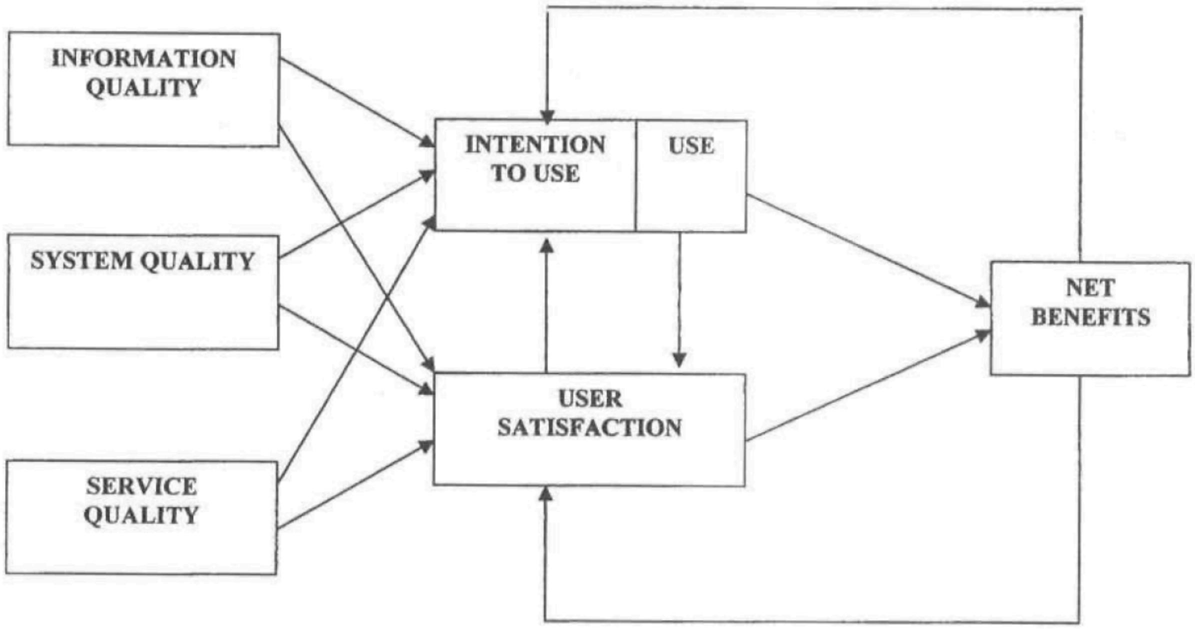


Figure 3 D&M IS Success Model

II.2 OSS project success measures

OSS is a unique type of system development, and it differs vastly from traditional software development practices. Proprietary projects are developed in a structured environment with a pre-determined set of resources and controls (See Table 3). Unlike these projects, the public can download the computer program of the software in OSS projects (Sen et al. 2011). The latter projects are designed and developed through the voluntary contributions of developers. OSS projects extend beyond a single organization since a community of developers from different organizations build the software code. Later, Crowston et al. (2003) opined that the measures posted by Delone and Mclean are hard to justify for the OSS projects and proposed several measurable criteria (project output, success, and outcomes for project members) to serve as indicators for the success of OSS Projects. Crowston et al. (2003) and Subramanian et al.

(2009) concluded that any single measure could not be the final word on success and suggested using a portfolio of tests that draw on different perspectives for evaluating OSS Projects.

Table 3 OSS and Proprietary Applications

Applications	**Open Source Software**	**Proprietary**
ERP	Metasfresh	Oracle EBS
Browser	Mozilla Firefox	Microsoft Internet Explorer
Database	Oracle relational database	MongtoDB NoSQL Database
Office productivity suite	Microsoft Office	Apache OpenOffice

The existing literature has also provided different determinants for the success of open source projects. OSS literature has identified that voluntary contribution of the developers, capability to attract financial donation from major corporations, and the ability of users to modify the software contribute to the success of OSS projects. The most recognized determinant of OSS success is the participation of developers in creating, developing, and maintaining the software (Ravi Sen et al. 2012).

Intellectual property rights (IPR) play a significant role in the IS projects and, more specifically, OSS projects. Owing to the significance of IPR in OSS projects, Wen et al. (2013) discovered that OSS projects with a high degree of overlap with disputed OSS exhibited a more significant decline in the adaptability of the software. The enforcement of IPR action on an OSS project significantly impacts its success.

The OSS projects are coded in multiple programming languages. Successful projects attract skilled developers, many users and company sponsorship. The developer base is referred to as the number of developers participating in a project in a given period, and the subscriber base is referred to as the number of peoples subscribing to an OSS project in a given period

(Stewart et al. 2006). Through an empirical study, Sen et al. (2011) discovered that the projects using a specific programming language such as *C* or its derivative exhibited a higher degree of subscriber base than the projects lacking these characteristics. Another key finding of this research is that OSS projects with restrictive licenses attracted fewer subscribers and developers (Sen et al. 2011). The study also concluded that the influence of subscribers and developer base increased with the age of the project.

Although knowledge sharing is a vital component of a successful OSS project, it also requires the developer's attention to be successful. As software developers participate in multiple projects, their attention towards the focal project diminishes, thereby lowering the chances of its success. Daniel et al. (2016) explored how knowledge integration, developer attention and network degree centrality influence the success of OSS projects.

The research on OSS project success has a profound influence on software managers and project administrators. The longitudinal study performed by Subramaniam et al. (2008) indicated that restrictive OSS license has a negative impact on the success of OSS projects. Also, the study identified that the success measures of activity levels, user interests, and developer interests are interrelated to one another. The data for this study primarily comes from the open source projects hosted at Sourceforget.net. However, the OSS success studies failed to consider the social collaboration and the social factors involved in the creation of the project.

II.3 Network formation

The study of social factors that constitute the OSS project team and its impact on the success of the project provides a set of recommendations for OSS project managers to follow. The collaborative social model offers a collection of new templates that improve the software development process. However, challenges exist in the collaborative structures that impact the success of the OSS project. Rajdeep et al. (2006) argued that open source systems need to be viewed as a network and that the project managers with a high degree of social capital will be able to create teams with technically diverse skillsets (Ruef et al. 2003). The study also identified that network embeddedness has substantial effects on both the technical and commercial success of OSS projects (Rajdeep et al. 2006). Network embeddedness depicts the variations in the network ties, and the study explores the relationship between the heterogeneity of social capital and network embeddedness in the success of open source projects.

The OSS environment is characterized by a set of developer volunteers having the common objective of developing a software product. A successful open source project involves building a team of talented resources. The companies working on an OSS project can hire a project founder; however, the projects cannot succeed based solely on the project founder and their social capital. The social capital of the project founders is determined by the size of the team and team brokerage. The study by Wang et al. (2018) concluded that the size of the team and team brokerage contribute differently to the success of OSS projects.

The open source project thrives on knowledge sharing across developers and projects. The project is created by the developer in a repository such as Github, SourceForge or Bitbucket. Subsequently, the OSS framework allows additional developers to modify the source code and provide other features and enhancements. The knowledge gained from one project can be applied to additional projects. As the promotion of knowledge sharing is a critical component of the OSS framework, Singh et al. (2011) discovered that the projects with greater internal cohesion, moderate levels of external cohesion, and technological diversity of the external network have a higher success rate. As the projects are virtual in nature, communication becomes increasingly difficult and a high degree of internal cohesion provides trust and better knowledge sharing among the team members. The open source developer network proposed by the study is illustrated in Figure 4.

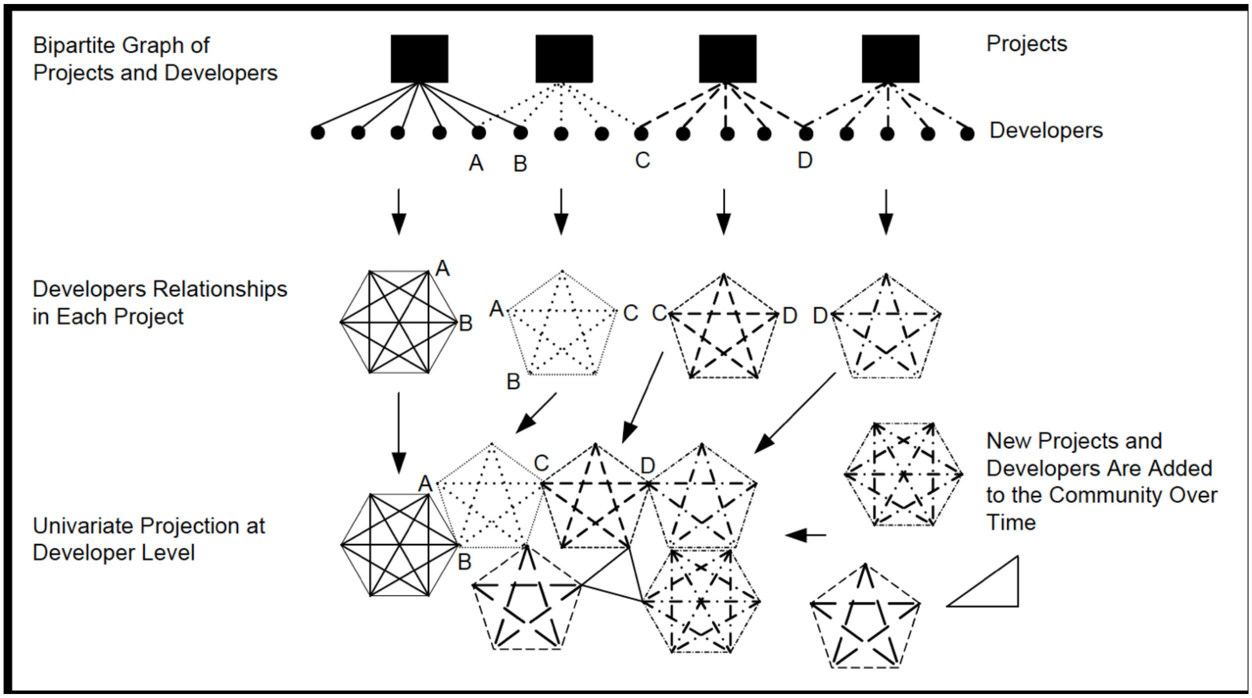


Figure 4 Open Source Developer Collaboration network (Singh et al. 2011)

The decentralized open source ecosystem requires a better understanding of the OSS community. The developers and users forge a sense of relationship, and several studies have explained the OSS network phenomenon. The empirical research conducted by Madey et al. (2002) revealed that the OSS community is formed through a self-organizing developer network. The results revealed that the developer's attachment to the project is not a random phenomenon; it rather occurs due to the existing ties between the feedbacks of the developer(s) on the projects. The study defined a set of software developers to be connected if they are members of the same project or if they are linked through a chain of related developers (Madey et al. 2002).

II.4 Online Developer Communities

Sociologists have studied the phenomenon of new team formation, and such studies have provided a macro-level view of the group formation concept. Research by Ruef et al. (2003) concluded that homophily, strong ties and isolation have a profound influence on the formation and composition of the teams.

The open source collaboration network can be described as an affiliation network. It is represented by the affiliation between two groups – one group representing the development and another denoting the activities performed by the developer in the OSS environment. The developers are related to each other through activities such as code development and testing performed by them (Wasserman et al. 1994). A developer working on two or more open source project form an affiliation network. Such an affiliation network for an open source project is provided in Figure 5 (Singh et al. 2007).

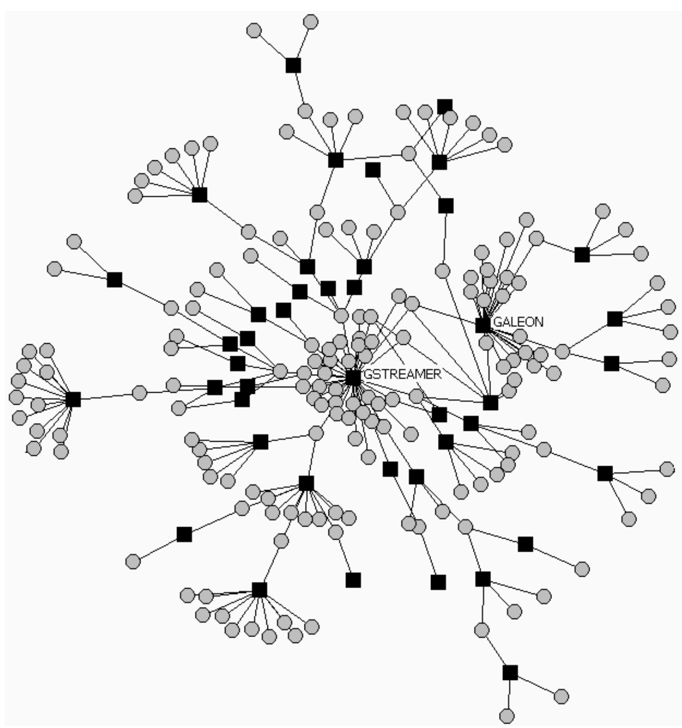


Figure 5 Affiliation network for Gnome foundry

Black squares represent projects, and grey spheres represent developers.

The open source projects are developed by a pool of software developers, and the OSS communities evolve over time. The presence of an OSS project in the repository alone is not enough to make it successful. Well-established companies and enterprises use a community manager to promote open source projects and attract developers. The study performed by Jiang et al. (2016) concluded that the size and diversity of the developer community affect the productivity of the open source community.

Software development involves several challenges and requires theoretical and practical knowledge (Sacks 1994). The knowledge gained by resolving one issue can be applied to similar problems in another project. The difficulty is also related to how one of the solutions can be applied to resolve the issue (Boh et al. 2007). The informal knowledge to identify and address the issues through the best solution is kept within the developers, and gaining access to this knowledge will enable a better design and quicker resolution to the issues (Singh et al. 2007).

The social media and the internet, through knowledge sharing, have provided answers to several questions. The community-based knowledge sharing domains have become popular over the last decade. Social interactions between the developers have significantly increased through the community portal Stack Overflow (Blanco et al. 2019). Such communications are crucial to knowledge sharing. Chou et al. (2010) discovered that collaborative elaboration and communication competence impact the completion of OSS project tasks. However, the literature has not addressed how new teams emerge from the online developer community and whether they are successful. Figure 7 provides a view of how the developers got voted in questions.

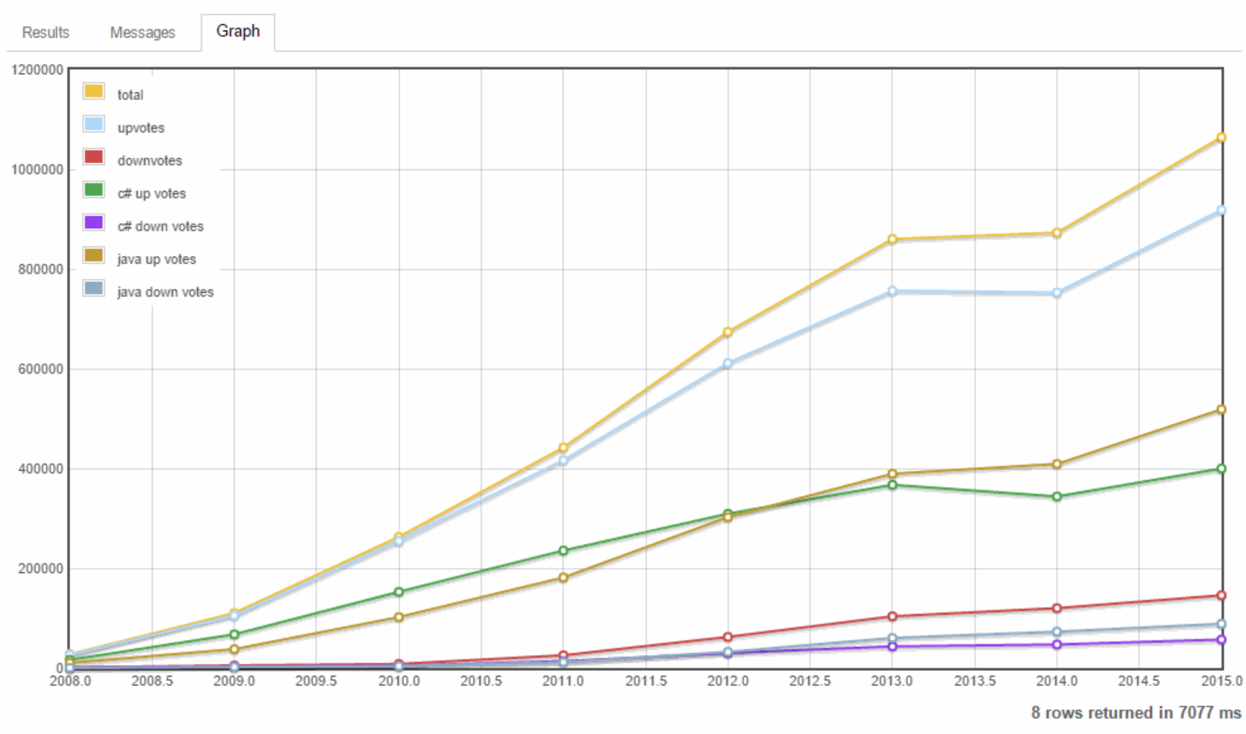


Figure 6 Stack Overflow trend

II.5 Developer Sentiment

Social media databases hold the opinions of millions of users. Individuals post their views on a social media website, and the advent of mobile technology has eliminated the constraints in the posting of opinions (Deng et al. 2018). Open source repositories contain the feedback from users in the form of opinions and comments. Sentiment analysis refers to the process of analyzing the input and ideas in textual format and categorizing them as positive, neutral and negative sentiments for decision making. The research performed by Ikram et al. (2015) indicated that the adaptability of OSS products increases with positive sentiment. The current literature has not addressed the relationship between developer sentiment and the success of open source projects.

II.6 Literature gap

While the existing research has identified key OSS success measures and determinants, they have failed to recognize the factors in the context of OSS new feature requests and patch requests, an exception being the work by Temizkan et al. (2015). This study was performed on projects using a single programming language (Programming Language "C") as the network boundary and relying on network social capital as the success factor. The developers with different forms of technology skills (technological diversity) are prone to produce new knowledge and improve the reliability of the OSS product. The data in this study was confined to projects using the programming language. Besides, the work did not include the technological diversities and the propensities of enterprise firms. Also, the data in most of the literature are based on a collection of open source projects in the repository "SourceForge.net". Furthermore, the studies failed to consider the other prominent secondary open source project data source "GitHub", which hosts a variety of open source projects that vary greatly in size, number of developers, and programming languages.

Reusable software codes are abundantly available in the open source libraries, and access to diverse technological resources increases the number of innovative solutions to technical problems. Prior studies on large OSS projects such as Linux and Apache (Bergquist et al. 2001) have demonstrated the contribution of network social capital factors. Given that such elements influence OSS projects, the characteristics of the developers, such as technological diversity, are also likely to affect the success of the projects. Three network social capital entities (Table 4) and two moderating entities (Table 4) that can impact the success of OSS projects have been identified in this study. The control variables are technological diversity, age of the project, and size of the project teams.

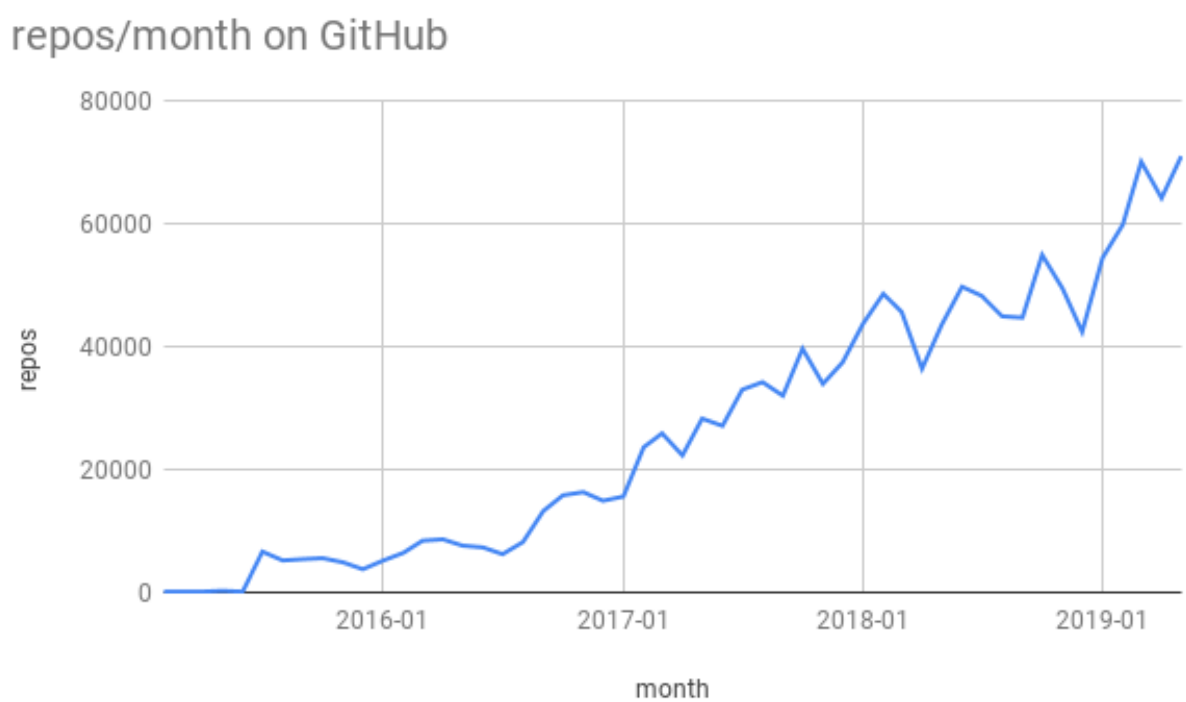
Table 4 Key Constructs

Constructs	**Description of constructs**
Technological Diversity	Characteristics of the individual having a knowledge of diverse technologies
Internal Cohesion	The degree to which internal project members collaborate with each other
External Cohesion	The degree to which external project members work with each other
Artifact type	Type of request – patch request, feature request
Patch Request	Requests to correct the faults in the existing software
Feature Request	Requests to add new features to the existing software or add new software modules based on user requirements

The researchers have primarily studied the social network perspective of open source software development through the SourceForge project community. The phenomenon of group formation is largely studied through ties among the developers in the OSS project community. The current literature does not address the group formation relationship between a distinct developer community such as stack overflow and a project community such as Github. Besides, the current literature does not address the success of open source projects and how a new project team is formed through an online developer community. The present literature also lacks a study of developer sentiment and project success in the context of highly enriched Github and Stackoverflow platforms.

Given the gaps in literature, the impact of online developer community, Stack overflow, and developer sentiment on the success of the project was researched in this study. Moreover, the influence of artifact type on the relationship between the participation level of stack overflow developers and OSS project success was also investigated. As the prior studies focused only on a single programming language and “SourceForge” project datasets, this work employed multiple programming languages as the network boundary and also made use of the open source project data foundry “GitHub.” Figure 7 presents the growth of the GitHub repositories over the last three years.

Figure 7 Github Growth



III THEORETICAL FRAMEWORK AND HYPOTHESES

III.1 Theoretical Background

Open source software is described as a “collective invention” (Nuvolari, 2005) in which developers freely share their expertise to produce new knowledge and products. Software products are often developed using a modular approach to design new modules as well as make improvements to the existing ones through innovative solutions (Sacks 1994). In addition to inheriting, integrating and making modifications to the current code, the developers gain adequate troubleshooting skills by engaging in software development (Boh et al. 2007). The success of an information systems project depends on the team’s ability to generate knowledge and transfer it within and across the boundaries (Ayas 1996). The knowledge gained on a software project can be applied to develop solutions for a similar project and reduce the delivery time (Singh et al. 2010). An open source developer can work on multiple concurrent software projects, and the knowledge gained can be effectively applied for related projects.

The open source developers and users form a complex social network of relationships through electronic communication channels (Hippel et al. 2003). Social network is based on graph theory, which postulates that a network can be designed in the form of a graph, with developers representing the nodes and the connections among the developers denoting the edges (Wasserman. 1999). The collaborative networks are an offshoot of a social network in which the connections between the developers are collaborative in nature (Madey et al. 2002). The open source developers form collaborative relationships with others in an open source project community such as Github or with an entirely different developer community such as Stack Overflow. Previous research suggests that the strength of the relationship depends on variables such as length of the relationship, emotional intensity, and reciprocal engagement related to the relationship (Granovetter 1973). The strength of relationship between the developers within the community plays a crucial role in the creation of an open source project team. In this paper, we focus on the prior collaborative relationship between developers in the stack overflow community as a driver behind the creation of open source project teams. The engagement of the community in open source projects and its influence on the success of these projects has also been examined.

Various researches have reiterated that successful organizations are ambidextrous. A study by Newbert (2007) demonstrated the importance of resources in the performance of an organization. Ambidextrous firms acquire a competitive advantage through exploratory and exploitative innovations (Benner and Tushman, 2003). Organizational literature identifies three critical categories of ambidextrous process capabilities, namely, structural, contextual and leadership (Raisch and Birkinshaw 2008). Structural antecedents relate to the structural mechanisms that are implemented to balance the tradeoffs faced by the organization. Contextual antecedents are associated with the systems and processes that are deployed to balance the conflicting demands of an organization. (Lee et al. 2006). Leadership antecedents are linked to the leadership qualities required to support organizational ambidexterity.

The ambidextrous organizations will be able to reap a better success by balancing both the exploitative and exploratory initiatives and not preferring one over the other. Organizational learning theory indicates that the survival and success of any organization depend on the teams' and the firm's ability to aid in the exploration of new initiatives and the exploitation of old certainties (March 1991, Holland,1975). The exploitation and exploration framework considers two views on organizational learning involving the development and use of knowledge: the exploitation of existing resources and the exploration of new options. Exploring new initiatives is future-looking and involves various experiments (March 1991). Exploration is associated with novel ways of thinking and is captured by parameters such as variation, flexibility, discovery, and innovation. It is closely related to innovative ideas that completely change the trajectory of the used technology, besides significantly impacting the organizational competency. Exploration results in innovative designs and requires unique knowledge or departure from the existing one.

In the context of IT, a different set of organizational structures enables the exploratory team to produce innovative solutions and the exploitative team to develop the required solutions for the project. The development of a software product involves a set of activities that are related to adding new features (FR- feature requests) and fixing the issues with the existing product (PR – patch requests). Exploration is associated with experimentation, discovery, and risk-taking behavior (Choi et al. 2018). These activities are closely related to feature requests. Hence, it is suggested that such requests be called exploration activities. In contrast, the exploitation of software products refines the existing features of products through the implementation of patches. Hence, it is suggested that patch requests be called exploitation activities.

Exploiting the existing products is associated with variance-reducing activities (Farjoun 2010) via focus and refinement. Organizations have demonstrated that they can improve their teams and achieve high knowledge levels if they cultivate heterogeneous knowledge (March 1991). This research also indicated that "the essence of exploitation is the refinement and extension of existing competencies, technologies, and paradigm. The essence of exploration is experimentation with new alternatives" (March 1991). Firms can engage in different degrees of exploitation and exploration activities.

Such activities create incompatible and inconsistent actions (March 1991). Exploration instills a broad range of new and undeveloped ideas; in contrast, exploitation presents a narrow range of in-depth solutions. The former is associated with innovation, flexibility, and decentralization, and in comparison, the latter is related to efficiency, centralization, and refinement. This study specifically focusses on the impact of stack overflow community participation on patch and feature request associated activities.

The online communities thrive on knowledge sharing between the individuals and groups in the community. The knowledge sharing process is defined as the involvement of members who contribute knowledge and explore it for reuse (Chen et al. 2010). The developers from different backgrounds share their technical and professional knowledge with others in the community. The individual's self-motivation, interpersonal skills and organizational context play a major role in knowledge sharing among the members. The social exchange theory is well suited to explain this concept (Blau 1964). The developers are self-motivated to share their knowledge with the community, and the worthiness of the community depends on the quality of knowledge shared in the network (Chen 2007). According to the social exchange theory, a donor and a receiver are involved in a knowledge-sharing transaction. The donor determines what to exchange with the receiver. The members in an online developer community can exchange their knowledge to troubleshoot issues and guide other members in developing a new functionality. The OSS projects have a greater chance of success if there is a higher degree of knowledge sharing between the team members, which promotes innovation (Wang et al. 2012). The online communities such as stack overflow provide a forum to foster innovation through knowledge sharing between the members. In this study, we focus on the ties forged through knowledge sharing in the developer community and its impact on the success of the project.

III.2 Hypotheses

Research Question:

RQ1: Does the participation of Stack Overflow community developers influence the success of open source projects?

RQ2: How does the level of participation of stack overflow community developers impact the success of open source projects?

Hypothesis H1: *The greater the participation of stack overflow developers, the higher the success of open source projects.*

Open source software has evolved over the years, and they vary significantly in their technological composition and architecture. Knowledge is generated through variations in existing and new knowledge (Kogut et al. 1992). The team members with different technological expertise facilitate various forms of technical knowledge, capabilities, and alternative solutions. This approach fosters new thoughts, ideas, and innovative solutions to the existing problems (Sampson, 2007). The knowledge shared across a project team having diverse technical expertise is highly beneficial for the successful completion of the open source project. The repository Github provides a platform for the developers to publish their code and the project. A collaborative platform such as Stack overflow, enables the developers to share their skills and assist others. The developers share their knowledge when developing a code in GitHub and answering the questions in Stack overflow. Based on these arguments, a positive linear relationship is hypothesized between the participation of stack overflow developers and project success.

Hypothesis H2: *The more the reputation level of stack overflow developers, the higher the success of open source projects.*

The stack overflow site is focused on providing a forum to pose and respond to programming level questions. The developers offering high quality and highly ranked answers to questions and actively participating in discussions receive reputation points in the platform. The score measures the developer's activity and the quality of that activity in the network (Macleod, 2014). It could be inferred that a high reputation score implies the ability of the developer to share their high-quality talent with the rest of the community. Hence, it could be argued that a linear relationship exists between the reputation level of the stack overflow developers participating in open source projects and the success of the projects.

Hypothesis H3: *The higher the number of existing ties between the stack overflow developers involved in open source software projects, the higher the success of the projects.*

Sociology literature has proposed that the perceived status of any human being is related to their relationship with others (Frank,1985). The status of a relationship is based on the number of prior ties (Podolny,1993). A virtual community of developers builds open source projects. In this context, prior connections provide an opportunity to develop high-quality software as previous collaboration opportunities enable the sharing and gain of technical knowledge. Earlier collaborative ties also allow the project team to gain additional resources and increase the possibility of success. Hence, it is proposed that the existing developer ties in the stack overflow community positively influence the participation of the developers in the community.

Hypothesis H4: *The artifact type positively moderates the relationship between the participation of stack overflow developers and the success of open source projects.*

The feature-request teams capitalize on technological diversity, and they require new knowledge from different technical areas. However, the patch-request teams thrive on existing expertise, and they are focused on correcting problems with the existing open source software. The stack overflow developer community has extensive experience in providing answers to complex programming questions and in resolving code bugs. Thus, it could be stated that the participation of stack overflow developers has a differential impact on the success of patch and feature request teams moderated by the artifact type.

Research Question

RQ3: Does developer sentiment towards open source projects influence the success of the projects?

RQ4: Do both positive and negative sentiments influence the success of open source projects in the same way?

Hypothesis H5: There is a difference between the predictive performance of an open source project success model with sentiment and a model without sentiment.

Hypothesis H6: There is a difference between the predictive performance of positive and negative sentiment postings.

Open source projects need continuous and long-term participation from the developers. The socialization behavior of developers contributes to their long-term participation in the project (Qureshi et al. 2011). A co-evolution relationship exists between the open source software development coding practice and communities (Lindberg 2013). The general feeling about a project is reflective of the sentiment of the participants and end-users. The succinctness of the feedback facilitates the diffusion of information in the community. An open source project with positive feedback attracts more developers. On the other hand, repositories with negative feedback may make the developers abandon their participation. Hence, it is suggested that the sentiment towards open source projects has a predictive power on their success.

The developers participate in an open source project for different reasons (Robinson et al. 2016). The positive sentiment towards the project attracts additional talent from the community. When the projects receive negative reviews, the developers may decide to leave it. The negative sentiment reflects a lack of functionality or poor reliability of the product. The research on behavior finance indicates that investors react to good news and bad news differently (Barberis et al. 1998). Particularly, the investors respond more strongly to bad news than good news. Hence, it is opined that the predictive performance for postings of negative sentiment is higher than that of positive sentiment.

IV RESEARCH DESIGN AND METHODOLOGY

IV.1 Research Design

A cross-sectional quantitative research design was implemented to validate the statistically significant relationship among the participation level of the stack overflow developers, existing ties between them, and their reputation level by moderating the behavior of artifact type, developer sentiment, and the open source project success factor. Relevant data were collected from the raw secondary data available through the Google BigQuery database to test the hypotheses of the relationship among the various independent variables and the dependent variable, project success.

IV.2 Data Collection

IV.2.1 BigQuery Database

BigQuery is a serverless large-scale data warehouse developed and hosted by Google. The platform stores massive datasets containing useful information from various sources. Through its strong query engine, the database allows the users to conduct interactive querying and data analysis. BigQuery enables one to run a query that spans millions of rows and returns the results in seconds or minutes. The architecture allows the platform to be limited only by its infrastructure capacity. It also provides a robust Extract, Load and Transform (ELT) workflow, which is summarized in Figure 8.

Workflow	Architecture	When you'd use it
EL	Extract data from files on Google Cloud Storage. Load it into BigQuery's native storage. You can trigger this from Cloud Composer, Cloud Functions, or scheduled queries.	Batch load of historical data. Scheduled periodic loads of log files (e.g., once a day).
ETL	Extract data from Pub/Sub, Google Cloud Storage, Cloud Spanner, Cloud SQL, etc. Transform the data using Cloud Dataflow. Have Dataflow pipeline write to BigQuery	When the raw data needs to be quality controlled, transformed, or enriched before being loaded into BigQuery. When the data loading needs to happen continuously, i.e., if the use case requires streaming. When you want to integrate with continuous integration/ continuous delivery (CI/CD) systems and perform unit testing on all components.
ELT	Extract data from files in Google Cloud Storage. Store data in close-to-raw format in BigQuery. Transform the data on the fly using BigQuery views.	Experimental datasets where you are not yet sure what kinds of transformations are needed to make the data usable. Any production dataset where the transformation can be expressed in SQL.

Figure 8 Summary of the ELT workflow in Google BigQuery

IV.2.2 Github Archive BigQuery Database

The OSS project data for this research were gathered from the Github database (https://github.com/), which is a public software code repository. The developers create the code and synchronize the changes in the Github repository. The software developers use pull requests and issues to modify and enhance the software code to resolve issues and add new features to the project. Github provides 20 different event types that record the developer activities such as forking the repository, committing the code base for changes, and performing pull requests.

Github archive (GH archive) and GHTorrent databases are available publicly for research purposes. While the former stores the Github event stream, the latter stores them in a relational database for easy query access (Baltes et al. 2018). GH Archive stores the public data available in the GitHub project repository. The database contains GitHub project-related information from 2011 to date and is summarized in the form of daily, monthly and yearly tables.

Github provides REST APIs for researchers to mine the repositories and gather data. However, the APIs to research the entire dataset in a meaningful way are limited. The objective of the proposed work is to extend the prior research by analyzing the rich source of relational data offered by Github archive and finding the factors that determine the success of OSS projects in the platform. The schema of the database is provided in Figure 9.

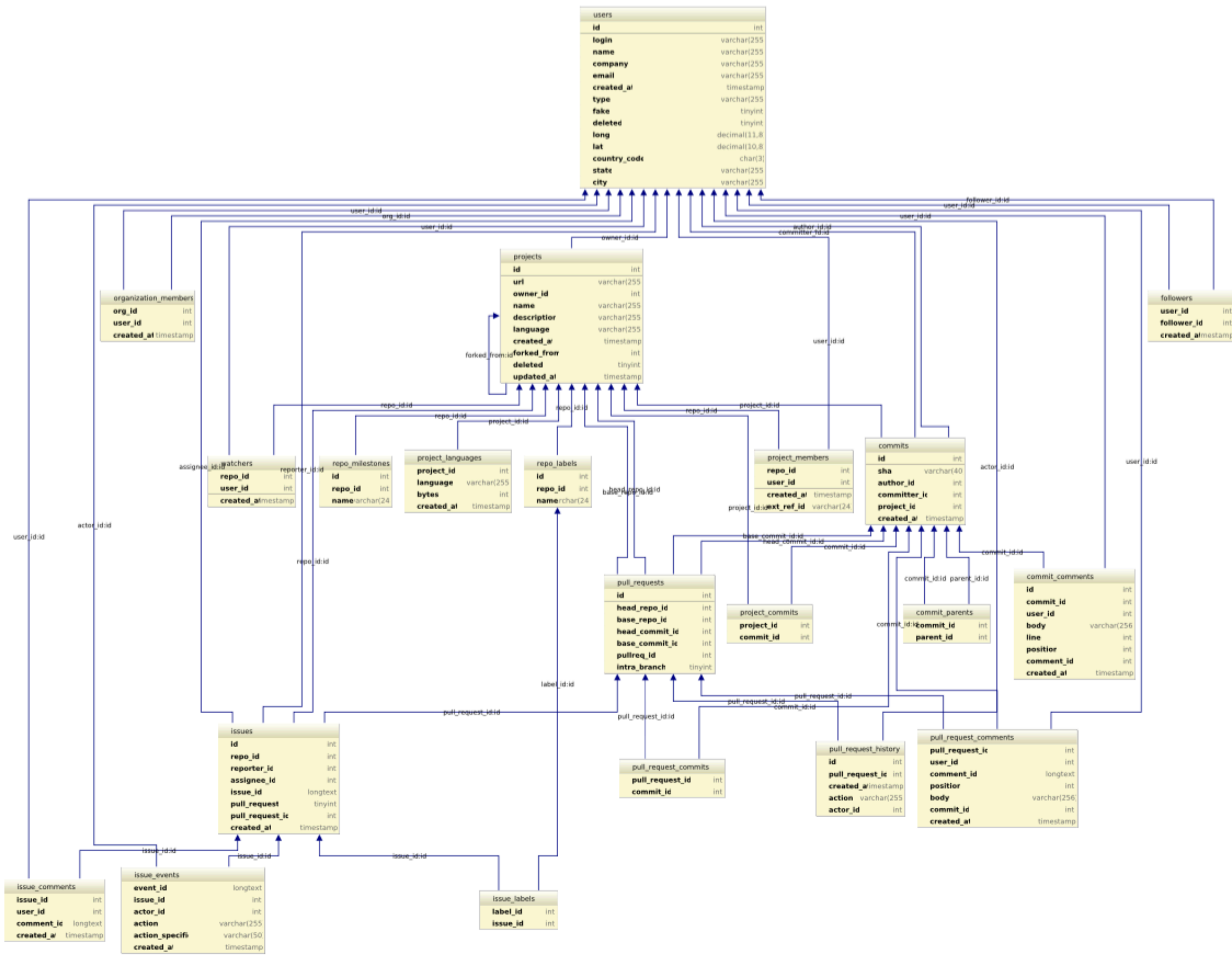


Figure 9 Github database Schema

The average volume of the GitHub archive datasets is summarized in Table 5.

Table 5 Summary of GitHub archive datasets

Github Archive Tables	Average Table Size	Average Number of Table Rows
DAY	3 GB	1.29 M
MONTH	175 GB	55 M
YEAR	1.68 TB	600 M

IV.2.3 Stack Overflow BigQuery Database

The stack overflow data were collected from the BigQuery database. The dataset is available publicly and updated every quarter (See Table 6). It contains various details about the stack overflow community such as posts, votes, comments, answers and badges. The schema of the database is shown in Figure 10.

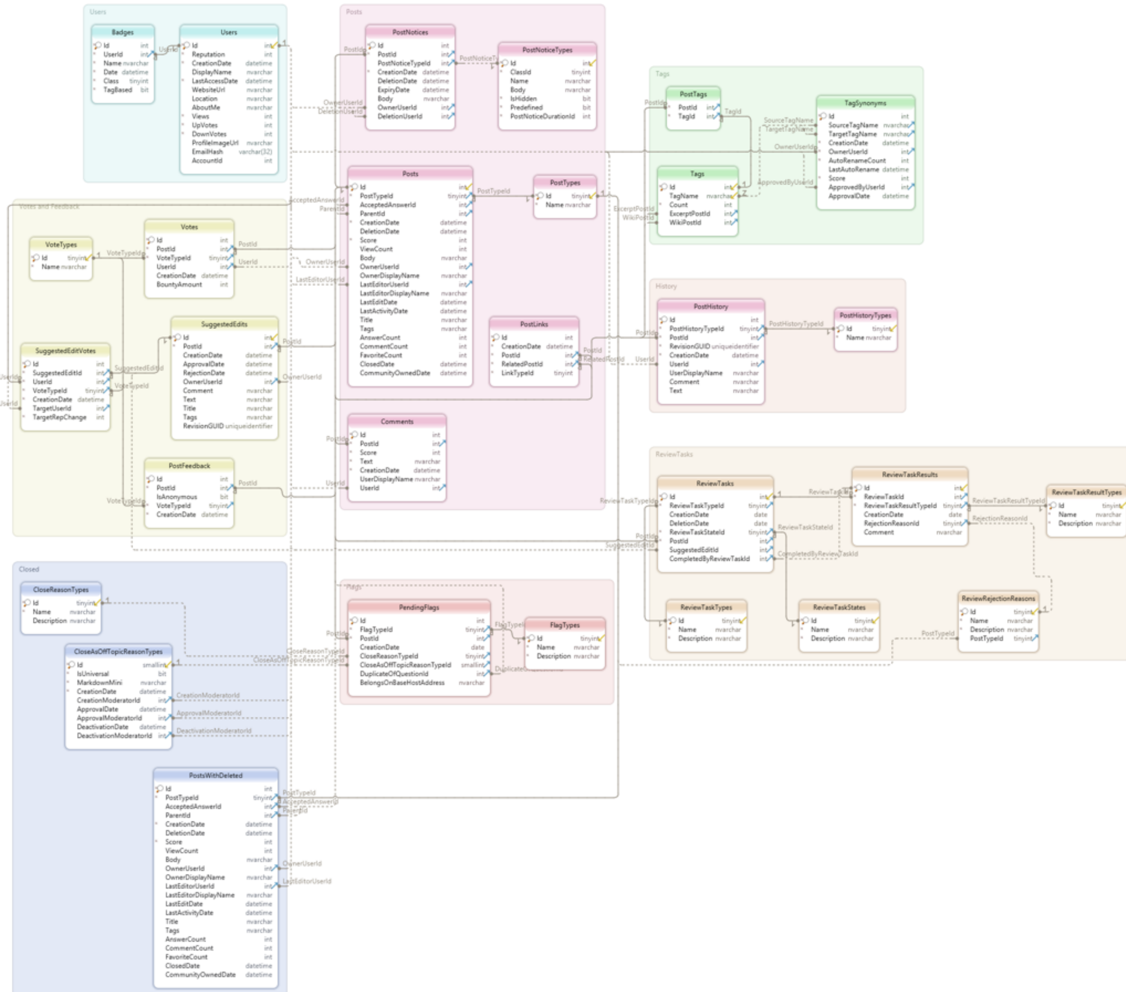


Figure 10 Stack Overflow Schema

Table 6 Summary of stack overflow datasets

Stack Overflow Tables	**Average Table Size**	**Average Number of Table Rows**
Badges	1.5 GB	33M
Comments	13 GB	74M
Post_history	85 GB	120M
Post_links	239 MB	6M
Users	1.71GB	11M
Votes	5.44GB	182M

For this research, the experimental setting was chosen as the hypotheses must be formulated, tested, and evaluated once formed. This research examined two measures of project success: developer contribution and the number of programming languages used. Both extrinsic and intrinsic attributes were part of the research model. The data for this research came from the information on projects listed in GitHub and expressed in relationship data format in the BigQuery database. As of now, the database contains information on close to 95,540,347 projects. For this study, those projects performed between January and December 2019 were chosen. Grouping the projects based on evolution time helped in exploring how various independent variables impact a project's success in different stages. Several strategies such as queries were used to increase the internal validity of the findings in the sampling process for the data drawn from GitHub to measure a project’s success. During data collection for analysis, all counts were taken at the project level.

IV.3 Data Analysis

The following steps were performed to complete the data analysis (Figure 11).

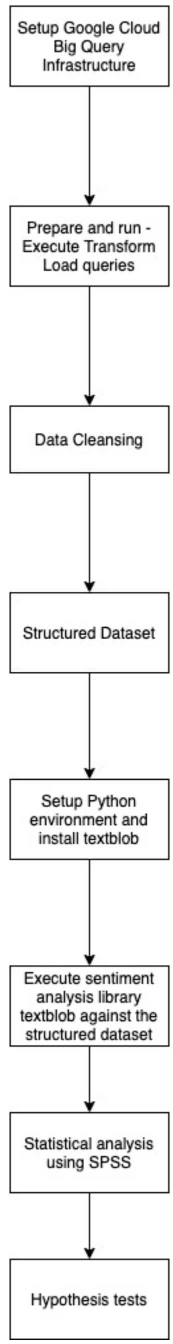


Figure 11 Data Analysis Process Flow

IV.3.1 BigQuery Infrastructure, ETL setup, and data cleansing

The research data from the BigQuery public datasets, namely Stack Overflow and GitHub, were used as the source database tables for analysis. The community developers participating in the open source project were identified by the name of the Github repository published in the profile. Data analysis involved extracting the raw data by cross-referencing the Stack Overflow user tables and the GitHub repo table with the help of the project name specified in the profile. An SQL query combining Stack Overflow and GitHub table was created, and the output of the query was stored in a separate dataset. The project names listed in the dataset were used to form another SQL query that extracted the necessary project details from the GitHub archive dataset. The output was merged with that of the first query to create the input dataset to the "data cleansing" process.

IV.3.2 Data Cleansing

The output of the ETL process was checked for errors and consistencies among the fields. Minimal and maximum values for relevant fields were reviewed and any inconsistencies were removed. The ties between the Stack Overflow developers created duplicate project data, which were used for validation but excluded during the SPSS statistical analysis phase.

IV.3.3 Sentiment analysis setup using Textblob

Textblob is a python library used for processing text data. The library provides an application programming interface (API) to perform sentiment analysis of textual data. Textblob offers a polarity score which ranges from -1 (most negative) to 1 (most positive). A python program was developed to perform sentiment analysis using "Textblob" on a CSV file. The output of the program yielded a CSV file, which had sentiment indicators of the following values: '0' – Neutral sentiment, '1' – Positive sentiment and '2' – Negative sentiment.

IV.4 RESEARCH MODEL

IV.4.1 Conceptual Research Model

The conceptual research model of this study is provided in Figure 12.

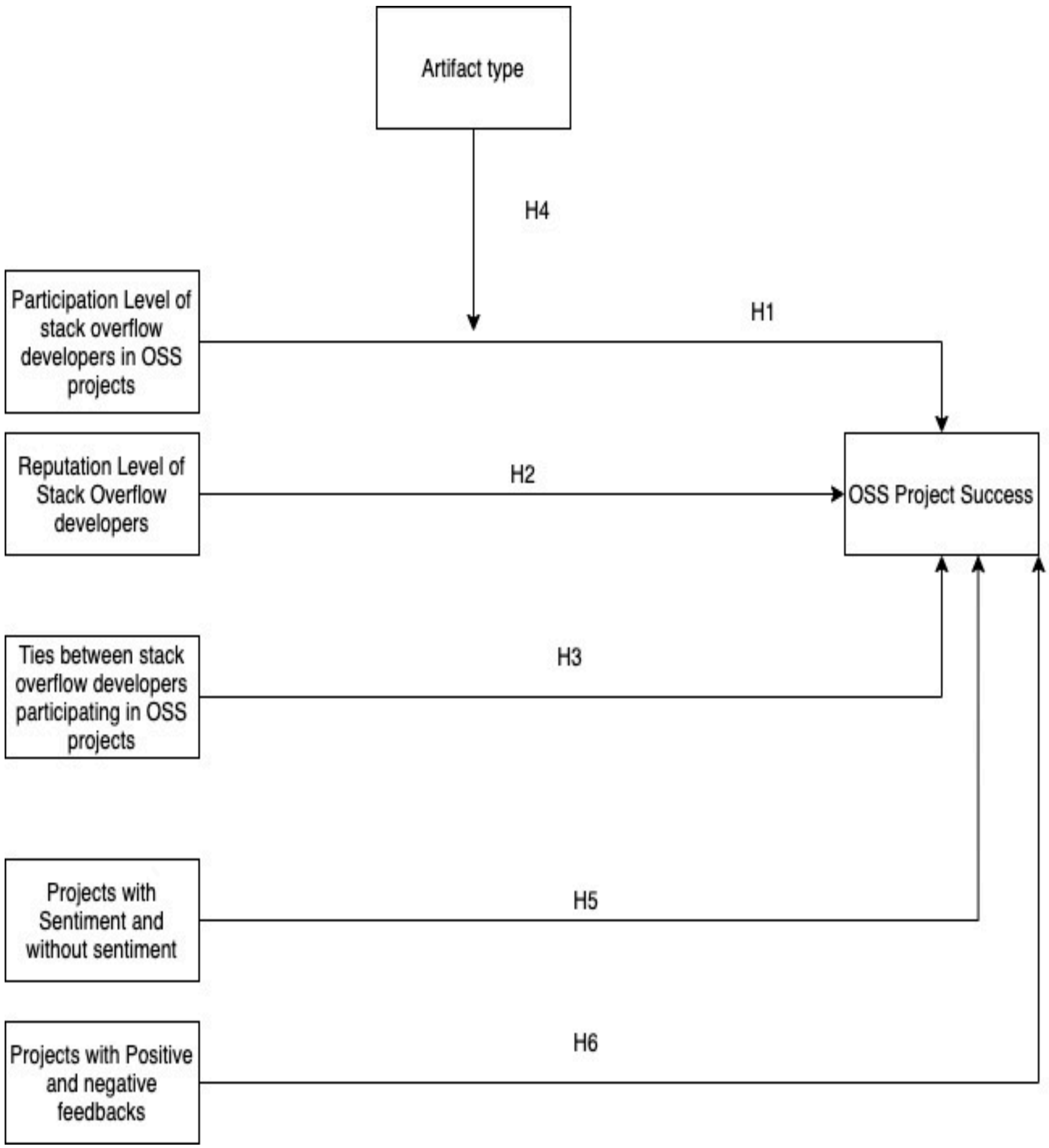


Figure 12 Research Model

IV.5 Dependent Variable

IV.5.1 Project success

Project success was taken as the dependent variable, and the number of commits was used as a measure of project success. The commit event happens when a developer loads a modified source software code into the project repository. As the event depicts changes in the source repository, the number of commits was portrayed as a measurable addition of functionality to the project. Several studies on OSS project success have utilized the number of commits as a determinant of open source project success. (Temizkan et al. 2015, Singh 2010, Crowston et al. 2003)

IV.6 4Independent Variables

IV.6.1 Control variables

Control variables were included in our research to account for the effect of factors other than the independent variables. The former depicts the characteristics that may cause differences in the dependent variable because of demographic issues, such as the age of a project, and activity level, such as the size of the team. The age of the project (in months) and the size of the team have been studied in the past as determinants of success and have been included in this research as control variables (Ravi Sen et al. 2012). In this work, the measure of technology diversity refers to the different programming languages used by the developers to build the open source software. The age of the project reflects the amount of dedication exhibited by the owners and the supporting team members to enhance the project. The study also included the number of languages used as a control variable.

IV.6.2 Moderator variable - Artifact type

In this research, the artifact type was used to control its moderating effect on the participation level of stack overflow developers in open source projects ((Temizkan et al. 2015). The artifact type was constructed with a value of 0 for feature requests and a value of 1 for patch development requests. The projects hosted in the open source repository create a variety of architects. The "feature request" artifacts reflect the number of enhancements and new features included in the open source project. In contrast, the "patch development request" artifacts depict the software code added to fix the bugs associated with the OSS project. The number of artifacts can vary based on the type and age of the project, and they also represent the changes done to it. The artifact type was derived at the project level from the OSS project repository Github.

IV.6.3 Participation Level

The developers often create a new repository by copying another one from the OSS project repository. The forking command is a built-in feature of the Github platform. The developers fork repositories to create new projects and add features and enhancements to them. An analysis of the forking phenomenon in the OSS project repository enables the project administrators to understand the OSS community, and more specifically, the participation level of the developers, which is construed at the OSS project level.

IV.6.4 Ties between the developers

The measure of ties between the developers could be defined as the result of existing collaborative relationships between the developers in the stack overflow community. The ties could be defined as those developers who have exchanged questions and answers in the developer community network and have participated in the same open source project. This relationship is identified by the presence of similar open source project repository names in the user profile of the developers. The self-organizing nature of the OSS teams allows the developers to join the projects at their own will. The developers may join the GitHub project community because of their existing relationship with the project administrator or other members of the team.

IV.6.5 Reputation Level

The stack overflow community has a rewards feature that enables the developers to gain additional privileges in the portal, such as site analytics and creating tags and chatroom. The reputation level is a numerical measure assigned by the platform for posting insightful questions and providing helpful answers to the community. The higher the reputation level of the developer, the higher the privileges received by them.

IV.6.6 Developer Sentiment

The developer sentiment is defined as the perception of the developers about an open source project. The concept is derived by accumulating the comments from the developers on various pull requests of the projects committed by the stack overflow community developers. The python library "Textblob" is used to mine the text data and provide the sentiment data.

IV.7 Statistical Analysis

A multiple linear regression analysis was performed to test for the presence of a correlational relationship between the selected stack overflow characteristics and their influence on open source project success. Project success was defined as the number of commits performed on the GitHub projects. Regression analysis was done using the SPSS software.

V RESULTS

V.1 Descriptive Statistics and Correlations

We used standard regression analysis to observe the influence of Stack Overflow community on OSS projects from Github. Initial investigation revealed that the dependent variable and a few of the independent variables were not normally distributed. Hence, the dependent and independent variables were logically transformed, and regression analysis was performed (Gelman et al. 2007). Table 3 reveals that the dependent variable (project commits) in this study has a mean of 482.86 and a standard deviation of 2677.957. The independent variable, participation level, has a mean of 48.87 and a standard deviation of 227.703. The reputation level has a mean of 1550.49 with a standard deviation of 7038.212. The control variable, age of the project, has a mean of 31.99 with a standard deviation of 26.09. The size of the projects has a mean of 2.41 with a standard deviation of 23.028, and the number of languages has a mean of 0.96 with a standard deviation of 2.204. The total number of samples used in the analysis was 758 (N=758), and these were collected over the entire year of 2019.

Table 7 Descriptive Statistics

	Mean	Std. Deviation	N
Project Success-commits	482.86	2677.957	721
Participation Level	48.87	227.703	705
Ties	.33	1.295	758
Age of the Project	31.99956 0246262075	26.09739 1290809792	758
Size of the Project	2.41	23.028	758
Number of Languages	.96	2.204	758
Reputation score	1550.49	7038.212	758

V.2 Regression Model Summary

The significance of the model was tested using the p-value. As shown in Table 8, the p-value was significant at 0.05 level. The R^2 value of the model was 0.142, which indicates that the model explains 14.2% of the relationship and is a reasonable fit. The coefficients of sentiment analysis are given in Table 9.

Table 8 Model Results (Dependent variable: Number of commits, N = 758, Coefficient Matrix)

Variable Name	Project Success			
	Model1	Model2.	Model3	Model4
Participation Level	0.214*	0.162*	0.159*	0.151*
Ties between stack overflow developers	0.111*	0.112*	0.115*	0.112*
Age of Project team	0.766*	0.654*	0.634*	0.652*
Size of Project Team		-0.055	-0.052	-0.064
Number of Languages used in the project		0.183*	0.181*	0.163*
Reputation score of stack overflow developers			0.030	0.028
Participation Level X Artifact Type				0.327*
				.
Sentiment				0.882*
R^2	0.901	0.909	0.909	0.910

*Significant at the 5 percent level

The general model could be represented using the following equation:

$Y = \beta_0 + \beta_{Pd} P_d + \beta_{Fp} Fp + \beta_{sent} F_{sent} + \beta_3 M_{at} P_d + \beta_{age} F_{age} + \beta_{size} F_{size} + \beta_{lang} F_{lang}$

Y = Dependent variable – Number of Commits

P_d = Independent variable – Participation Level of the stack overflow developers

Fp = Independent variable – Ties between the existing developers in the stack overflow community

F_{sent} = Independent variable – Sentiment Level of the stack overflow developers

F_{age} = Control Variable – Age of the project

F_{size} = Control Variable – Size of the project

F_{lang} = Control Variable – Number of languages used in the project

M_{at} = Moderator Variable – Artifact type

β_{Pd} = Coefficient relating the independent variable T_d to the dependent variable Project success - The effect of participation level of the Stack Overflow developers involved in the OSS projects on the number of commits

β_{Fp} = Coefficient relating the independent variable Fp to the dependent variable Project success- The effect of prior collaboration ties between the SO developers on the number of commits

β_{sent} = Coefficient relating the independent variable F_{sent} to the dependent variable

Project success - The effect of sentiment level of the SO developers participating in the OSS projects on the number of commits

β_3 = Coefficient relating the moderator variable M_{at} on the participation level of the SO developers (β_{Pd}) to the dependent variable

Project success - Moderating effect of artifact type to the participation level of the SO developers involved in the OSS projects on the number of commits

β_{age} = Coefficient relating the independent variable Fage to the dependent variable Y- The effect of age of the project on the number of commits

β_{size} = Coefficient relating the independent variable F_{size} to the dependent variable Y- The effect of age of the project on the number of commits

β_{lang} = Coefficient relating the independent variable F_{lang} to the dependent variable Y- The effect of age of the project on the number of commits

Table 8 indicates the results of the regression model. It was found that the study confirms hypothesis 1 because the interaction of the participation level of Stack Overflow developers in the OSS projects from Github with project success is positive and significant (β=0.151, $p < 0.05$).

A significant relationship between the reputation level of SO developers and open source project success was hypothesized (hypothesis 2). However, the results did not support this hypothesis (β=- 0.028, $p > 0.05$).

A relationship between the existing ties among the developers in the SO community and open source project success was hypothesized (hypothesis 3). A significant coefficient (β=0.112, p <

0.05) was detected, which supports the hypothesis.

As stated, in hypothesis 4, the results supported the moderating impact of artifact type on the relationship between the participation level of the SO developers and project success. The coefficient for the interaction term was positive and insignificant (β=0.327, $p < 0.05$), a result which supports the hypothesis.

Furthermore, it was hypothesized (hypothesis 5) that a difference exists between the predictive performance of a model with sentiment and one without it. In support of this hypothesis, a differential impact was noted between the two groups of projects (β_1=0.274, β_2=0.801, $p < 0.05$).

In hypothesis 6, it was opined that a difference exists between the predictive performance of a model with negative sentiment and one with positive sentiment. In support of this hypothesis, a differential impact between the two groups of projects (β_1=0.724, β_2=0.395, $p < 0.05$) was noted.

Table 9 Sentiment Results (Dependent variable: Number of commits, N = 721)

Variable Name	Projects with sentiment	Projects without sentiment	Projects with positive sentiment	Projects with negative sentiment
Correlation between project success and participation Level	0.274*	0.801*	0.724*	0.395*

*Significant at 5% level

V.3 Summary

While not all the hypotheses were supported in our model, it is important to note that most of the independent variables influenced the OSS project success (See Table 10).

Table 10 Hypothesis Results

Variable Type	Hypothesis	Hypothesis Type	Tested Variable	Results
Participation Level of stack overflow developers	Hypothesis H1	Success	Number of forks in the Github repository	Supported
Reputation Level of stack overflow developers	Hypothesis H2	Success	Reputation score of stack overflow developers	Not Supported
Ties between stack overflow developers	Hypothesis H3	Differential Impact	Number of developers from the stack overflow community participating in the same Github project	Supported
Moderation impact of artifact type on the relationship between the participation of Stack overflow developers and the success of open source projects	Hypothesis H4	Moderator	Number of commits for a given project and the artifact type	Supported
Sentiment Analysis – Predictive performance of an open source project with and without sentiment	Hypothesis H5	Differential impact	Number of commits for a given project	Supported
Sentiment Analysis – Predictive performance of an open source project with positive and negative sentiments	Hypothesis H6	Differential impact	Number of commits for a given project	Supported

VI DISCUSSION

In this study, the impact of online developer community network on OSS projects was explored. The formation of new teams by those embedded in the online developer community network to create successful projects was investigated. The key results from this study are summarized in Table 11.

VI.1 Key Findings

Online developer collaboration network exerted an influence on the success of open source projects.

Studies on open source projects have demonstrated various factors that contribute to their success. This work was driven by the lack of research on the formation of self-organizing teams in an open source project environment. This study assessed the relationship between an exemplary online developer collaboration network, namely the Stack Overflow, and open source project success through the lens of social network theory. The results from the empirical study imply the positive influence of stack overflow developers on the success of open source projects. These findings suggest that when the stack overflow developers participate in an open source project, it is successful.

The critical component of OSS projects is its members. The study indicates that internal cohesion and the participation level of the stack overflow developers play a crucial role in the success of open source projects. The existing relationship between the developers carried over to the open source project community, and prior ties between them contributed to the success of the projects. A software developer is more likely to join a new open source project initiative if they have a strong collaborative relationship with the project initiator or other developers. Software development is a social network process that depends on a strong communication and coordination between the developers (Sawyer et al. 1998). The additional dimension of the type of artifact deployed in the open source project had a significant relationship with its success. Within the stack overflow developer community, the study did not see any connection between the reputation level of the developers and the success of open source projects.

The developer sentiment had an influence on the success of open source projects.

In this study, the impact of developer sentiment from the Stack Overflow community on the open source projects was investigated. The findings revealed that projects with sentiment showed a different level of success than those without it. In addition, the positive sentiment of the developers played a considerable role in the success of projects. The positive developer sentiment facilitated a significant level of watchers, which eventually led to the success of the projects.

Table 11 Findings and Contributions of this study

Determinants	**OSS success measure**	**Findings from this study**
1. Relationship between the SO community and OSS project success • Participation Level		
• Reputation Level	Project commits	Positive Impact
	Project commits	No impact

• Existing ties between the software developers • Moderating impact of artifact type	Project commits	Positive Impact
	Project commits	Positive Impact
2. Relationship between the SO developer sentiment and OSS project success • Predictive performance of an open source project with and without sentiment • Predictive performance of an open source project with positive and negative sentiments		
	Project commits	Differential Impact
	Project commits	Differential Impact

VI.2 Contributions

VI.2.1 Contributions to Academic Literature

The study has contributed to the extant theoretical literature on the formation of new software development teams in a virtual open source environment through the interactions between the developers in an online developer community network. The findings have also provided a perspective on how OSS projects attract new developers through the network.

This study has served as an empirical research in the context of stack overflow community and its impacts on the success of open source projects. Specifically, the work has explored the participation level of the developers and the internal cohesion among them in open source projects. The moderating effect of artifacts on the relationship between the stack overflow community and open source projects has never been studied in the past. This is a crucial finding as the stack overflow developers are proficient in problem-solving, and its impact on the success of open source projects is discernable.

Besides, the study has performed an empirical assessment of developer sentiment on open source projects. This facet has never been researched in the past and is therefore a key contribution to the literature.

The study has also added to the literature on the behavior of self-organizing teams in a collaborative environment through the lens of graph theory and online developer community.

The study, in general, has contributed to the literature on the determinants of open source project success.

VI.2.2 Contributions to Practice

The study can assist software development leaders, project managers and recruiting managers in understanding the contribution of developer collaboration network towards open source projects. This research has provided a framework for building a successful virtual software development team through the Stack Overflow community. The study has created an awareness among the leaders that a highly successful self-organizing virtual team can be built from the online developer community.

Furthermore, the study has enabled those who have been tasked with recruiting a highly talented open source project team for enterprises to specifically target the developers from the stack overflow community during the recruitment process. If permitted by the developer privacy options, the recruiters can aim at sending targeted emails to the highly talented stack overflow developers from a specific technology domain.

Moreover, the investigation has created a framework for the recruiting industry to build a software as a service platform for recruiting talented developers from the online developer collaboration network. The platform can learn from the problem-resolving capabilities of the developers and match their skills with the needs of the enterprises.

The findings also suggest that developers focused on joining an open source project should try to establish collaborative ties with others in the online developer community network.

VI.2.3 Limitations and Future Research

The quantitative research has described the power of an online developer community, such as stack overflow, on open source projects. A limitation of the study is the derivation of the relationship between the Stack Overflow developers and their presence in the open source projects. The work derives this connection only if the Stack overflow developers specifically mention the name of open source project in their profile. Hence, the investigation does not capture the multitude of developers who do not carry the project name in their profile. Hence, future research can be extended to identifying constructs that carry the relationship between the stack overflow developers and open source projects in Github.

Another limitation of this study is its cross-sectional design. This research has specifically analyzed the impact of the relationship over a single year. Hence, it can be expanded to assess the relationship over several years.

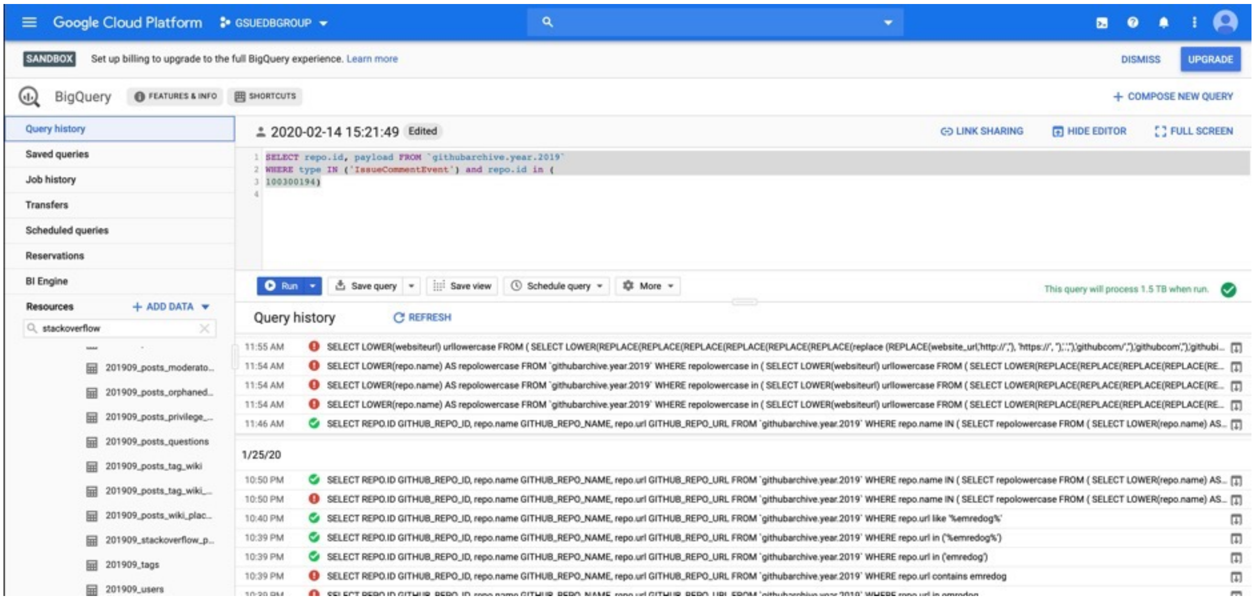
Besides, the study is also limited to the online developer community stack overflow and the open source project repository Github. Therefore, future research can be extended to additional developer communities such as “Experts-Exchange” and open source project repositories such as “Bitbucket”.

VI.2.4 Conclusions

In conclusion, the relationship between Stack Overflow developers and the success of open source projects was explored using the Social Network Theory as a theoretical framework. Our findings suggest that collaboration between the Stack Overflow developers results in a successful open source project. Additionally, the relationship between developer sentiment and open source project was examined. The open source projects with a high level of positive sentiment attracted additional involvement from the developer community and were successful. The recruiting industry needs to decipher ways to target skilled resources from the online developer community to build a successful project team. Such a community brings incremental value to a self-organizing virtual team, and future studies can include new developer communities and open source project repositories.

APPENDICES

Appendix A: Big Query Console



Appendix B: Table Pre summary

Appendix C: Table Post summary

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	Project Id	Comments	Classification	Confidence	stack overflow	GITREPONAME	GITTREPOURL	Age	CURRENT DATE	REPO_CREATED_AT	DESCRIPTION	REPO_FORKS	REPO_SIZE	WATCHERS_COUNT	REPO_LANG
2	96120	I'll send you	2	0.92	1	briot/geneapro	https://github.com/briot/geneapro	110	2/12/20	2/12/11		0	0	0	
3	248432		0	0.00	1	jceb/vimrc	https://github.com/jceb/vimrc	109	2/12/20	2/17/11	Vim configuration	8	1320	31	VimL
4	286061		0	0.00	1	sitaramc/gitolite	https://github.com/sitaramc/gitolite	110	2/12/20	2/12/11	Hosting git repos	159	392	2550	Perl
5	286061		0	0.00	1	sitaramc/gitolite	https://github.com/sitaramc/gitolite	110	2/12/20	2/14/11	Hosting git repos	159	392	2550	Perl
6	286061		0	0.00	1	sitaramc/gitolite	https://github.com/sitaramc/gitolite	110	2/12/20	2/13/11	Hosting git repos	178	1184	2727	Perl
7	286061		0	0.00	1	sitaramc/gitolite	https://github.com/sitaramc/gitolite	10	2/12/20	4/16/19	Hosting git repos	159	392	2550	Perl
8	286061		0	0.00	1	sitaramc/gitolite	https://github.com/sitaramc/gitolite	109	2/12/20	3/1/11	Hosting git repos	159	392	2550	Perl
9	286061		0	0.00	1	sitaramc/gitolite	https://github.com/sitaramc/gitolite	110	2/12/20	2/14/11	Hosting git repos	178	1184	2727	Perl
10	286061		0	0.00	1	sitaramc/gitolite	https://github.com/sitaramc/gitolite	110	2/12/20	2/12/11	Hosting git repos	178	1184	2727	Perl
11	286061		0	0.00	1	sitaramc/gitolite	https://github.com/sitaramc/gitolite	105	2/12/20	6/19/11	Hosting git repos	159	392	2550	Perl
12	286061		0	0.00	1	sitaramc/gitolite	https://github.com/sitaramc/gitolite	110	2/12/20	2/12/11	Hosting git repos	178	1184	2727	Perl
13	286061		0	0.00	1	sitaramc/gitolite	https://github.com/sitaramc/gitolite	110	2/12/20	2/13/11	Hosting git repos	178	1184	2727	Perl
14	296658	User","site_	2	0.60	1	haraldk/TwelveMonkeys	https://github.com/haraldk/TwelveMonkeys	109	2/12/20	2/22/11	The TwelveMonk	6	10968	9	Java
15	419430	I lost	2	0.83	1	yandod/candycane	https://github.com/yandod/candycane	109	2/12/20	3/15/11	a port of Redmin	37	10758	204	PHP
16	494735		0	0.00	1	maclennann/Athena-Chat	https://github.com/maclennann/Athena-Chat	106	2/12/20	5/26/11		0	0	0	
17	639740	When	2	0.65	1	sirthias/pegdown	https://github.com/sirthias/pegdown	106	2/12/20	6/14/11	A pure-Java Mar	17	413	119	Java
18	664467		0	0.00	1	adacosta/mongoid_rails_migrations	https://github.com/adacosta/mongoid_rails_migrations	104	2/12/20	7/21/11	Data migrations f	22	196	129	Ruby
19	780932		0	0.00	1	sunnygleason/j4-minimal	https://github.com/sunnygleason/j4-minimal	104	2/12/20	8/5/11		0	0	0	
20	992297		0	0.00	1	rmt/amqptools	https://github.com/rmt/amqptools	103	2/12/20	8/22/11	AMQP Tools - Cc	4	168	15	C
21	1140271		0	0.00	1	OpenRTMFP/Cumulus	https://github.com/OpenRTMFP/Cumulus	101	2/12/20	11/1/11	CumulusServer is	25	2740	193	C++
22	1254218		0	0.00	1	socketstream/socketstream	https://github.com/socketstream/socketstream	100	2/12/20	12/8/11	A fast, modular N	102	312	1835	JavaScript
23	1400125		0	0.00	1	ashima/webgl-noise	https://github.com/ashima/webgl-noise	99	2/12/20	12/16/11	Procedural Noise	10	2160	106	C
24	1482997	this issue	2	0.78	1	phpredis/phpredis	https://github.com/phpredis/phpredis	99	2/12/20	12/16/11		0	0	0	
25	1806785	would not	1	0.45	1	monora/rgl	https://github.com/monora/rgl	99	2/12/20	12/26/11		0	0	0	
26	1895389		0	0.00	1	autotelicum/Smooth-CoffeeScript	https://github.com/autotelicum/Smooth-CoffeeScript	99	2/12/20	1/2/12	A book on functic	34	360	457	CoffeeScript
27	1895389		0	0.00	1	autotelicum/Smooth-CoffeeScript	https://github.com/autotelicum/Smooth-CoffeeScript	98	2/12/20	1/29/12	A book on functic	34	360	457	CoffeeScript
28	2085509		0	0.00	1	yinkou/OCMapView	https://github.com/yinkou/OCMapView	97	2/12/20	3/7/12	Simple and easy	8	136	90	Objective-C
29	2161754	also	1	0.63	1	jsevellec/cassandra-unit	https://github.com/jsevellec/cassandra-unit	90	2/12/20	9/23/12	Utility tool to load	4	24154	19	Java
30	2250448		0	0.00	1	andythomas/TeX-sidenotes	https://github.com/andythomas/TeX-sidenotes	89	2/12/20	10/20/12		0	0	0	

VI.3 Appendix D: Table Post summary after Text analysis

1	Project Id	Comments	Classificatior	Confidence
2	96120	I'll send you	Negative	0.919
3	296658	User","site_	Negative	0.604
4	419430	I lost interes	Negative	0.831
5	639740	When runnin	Negative	0.648
6	1482997	this issue sh	Negative	0.779
7	1806785	would not ac	Neutral	0.452
8	2161754	also updated	Neutral	0.625
9	2993438	Thanks @ge	Positive	0.801
10	2996949	It's not just t	Negative	0.976
11	3644516	Referencing	Negative	0.796
12	3644516	Referencing	Negative	0.838
13	3644516	Referencing	Negative	0.796
14	4351207	f postman ca	Negative	0.875
15	4383703	No, I haven't	Positive	0.784
16	4604367	all the above	Negative	0.979
17	5099917	Pentesting fr	Negative	0.635
18	5887512	Hello, as this	Negative	0.738
19	5887512	Hello, as this	Negative	0.738
20	6025193	The option `-	Positive	0.855
21	6071776	YES,Working	Positive	0.536
22	7286816	Oops. Looks	Negative	0.927
23	8344254	Looks good +	Positive	0.676
24	8779606	thank you fo	Positive	0.413
25	8858625	awesome, th	Positive	0.947
26	11303194	Oh interestir	Positive	0.629
27	11309292	Ok, thanks. Y	Positive	0.836
28	11480313	I actually fix	Neutral	0.482
29	14022429	Confirmed, v	Negative	0.758
30	14699527	Hey, since w	Positive	0.626
31	15308499	This is an un	Positive	0.669
32	15763036	Fixed, thank	Positive	0.798
33	17113037	Thanks!	Positive	0.954
34	18920708	I'm sorry, thi	Negative	0.743
35	19248542	Work is ongc	Negative	0.996
36	20300177	We could co	Positive	0.696
37	21068958	"i want to cr	Neutral	0.878
38	21170833	Sadly, this te	Negative	0.98
39	23960944	Looks like fst	Neutral	0.46
40	24005390	You guys hac	Positive	0.767
41	24008590	You guys hac	Positive	0.767
42	24299860	You guys hac	Positive	0.551
43	25095239	the `lazyload	Negative	0.996

Appendix E: Model summary

Model Summary[f,g]

Model	R	R Square[b]	Adjusted R Square	Std. Error of the Estimate	Change Statistics				
					R Square Change	F Change	df1	df2	Sig. F Change
1	.949[a]	.901	.899	.62730	.901	683.924	2	152	.000
2	.953[c]	.909	.907	.60182	.009	14.058	1	151	.000
3	.954[d]	.909	.907	.60271	.000	.559	1	150	.456
4	.954[e]	.910	.907	.60131	.001	1.695	1	149	.195

Appendix F: Multiple Regression Analysis – Coefficients

Coefficients[a]

Model		Unstandardized Coefficients B	Unstandardized Coefficients Std. Error	Standardized Coefficients Beta	t	Sig.	95.0% Confidence Interval for B Lower Bound
1	(Constant)	-25.785	158.223		-.163	.871	-336.436
	Participation Level	1.243	.489	.106	2.540	.011	.282
	Ties	-196.457	88.055	-.095	-2.231	.026	-369.342
	Age of the Project	16.006	4.038	.156	3.964	.000	8.078
2	(Constant)	63.909	168.594		.379	.705	-267.106
	Participation Level	1.230	.492	.105	2.499	.013	.263
	Ties	-231.076	105.451	-.112	-2.191	.029	-438.117
	Age of the Project	9.861	5.252	.096	1.878	.061	-.450
	Size of the Project	-3.679	5.389	-.032	-.683	.495	-14.260
	Number of Languages	133.392	67.493	.110	1.976	.049	.877
3	(Constant)	72.998	169.112		.432	.666	-259.035
	Participation Level	1.232	.492	.105	2.503	.013	.265
	Ties	-238.561	105.986	-.115	-2.251	.025	-446.654
	Age of the Project	9.977	5.256	.097	1.898	.058	-.342
	Size of the Project	-3.765	5.392	-.032	-.698	.485	-14.352
	Number of Languages	139.673	68.064	.115	2.052	.041	6.036
	Reputation score	-.011	.014	-.028	-.728	.467	-.039
4	(Constant)	50.947	160.062		.318	.750	-263.319
	Participation Level	.957	.467	.081	2.050	.041	.041
	Ties	-232.061	100.306	-.112	-2.314	.021	-429.001
	Age of the Project	7.285	4.983	.071	1.462	.144	-2.500
	Size of the Project	-1.754	5.108	-.015	-.343	.731	-11.783
	Number of Languages	135.881	64.416	.112	2.109	.035	9.407
	Reputation score	-.009	.014	-.024	-.658	.511	-.036
	Moderator	15.694	1.734	.320	9.048	.000	12.288

Appendix G: Multiple Regression Analysis – Correlations

Correlations

		Project Success-commits	Participation Level	Ties	Age of the Project	Size of the Project
Pearson Correlation	Project Success-commits	1.000	.100	-.001	.151	.008
	Participation Level	.100	1.000	.436	.227	.344
	Ties	-.001	.436	1.000	.307	.576
	Age of the Project	.151	.227	.307	1.000	.300
	Size of the Project	.008	.344	.576	.300	1.000
	Number of Languages	.140	.300	.491	.681	.358
	Reputation score	.007	.007	-.039	.122	-.020
	Moderator	.337	.072	.021	.084	-.006
Sig. (1-tailed)	Project Success-commits	.	.004	.489	.000	.415
	Participation Level	.004	.	.000	.000	.000
	Ties	.489	.000	.	.000	.000
	Age of the Project	.000	.000	.000	.	.000
	Size of the Project	.415	.000	.000	.000	.
	Number of Languages	.000	.000	.000	.000	.000
	Reputation score	.430	.430	.143	.000	.287
	Moderator	.000	.029	.279	.010	.432
N	Project Success-commits	721	701	721	721	721
	Participation Level	701	705	705	705	705
	Ties	721	705	758	758	758
	Age of the Project	721	705	758	758	758
	Size of the Project	721	705	758	758	758
	Number of Languages	721	705	758	758	758
	Reputation score	721	705	758	758	758
	Moderator	721	705	758	758	758

Correlations

		Number of Languages	Reputation score	Moderator
Pearson Correlation	Project Success-commits	.140	.007	.337
	Participation Level	.300	.007	.072
	Ties	.491	-.039	.021
	Age of the Project	.681	.122	.084
	Size of the Project	.358	-.020	-.006
	Number of Languages	1.000	.144	.061
	Reputation score	.144	1.000	.001
	Moderator	.061	.001	1.000
Sig. (1-tailed)	Project Success-commits	.000	.430	.000
	Participation Level	.000	.430	.029
	Ties	.000	.143	.279
	Age of the Project	.000	.000	.010
	Size of the Project	.000	.287	.432
	Number of Languages	.	.000	.045
	Reputation score	.000	.	.490
	Moderator	.045	.490	.
N	Project Success-commits	721	721	721
	Participation Level	705	705	705
	Ties	758	758	758
	Age of the Project	758	758	758
	Size of the Project	758	758	758
	Number of Languages	758	758	758
	Reputation score	758	758	758
	Moderator	758	758	758

REFERENCES

1. Lindberg, A. (2013). Understanding change in open source communities: A co-evolutionary framework. In *Academy of Management Proceedings* (Vol. 2013, No. 1, p. 16619). Briarcliff Manor, NY 10510: Academy of Management.
2. Anthes, G. (2016). Open source software no longer optional. *Communications of the ACM*, *59*(8), 15-17. http://doi.org/10.1145/2949684.
3. Ayas, K. (1996). Professional project management: a shift towards learning and a knowledge creating structure. *International Journal of Project Management*, *14*(3), 131-136.
4. Baltes, S., Knack, J., Anastasiou, D., Tymann, R., & Diehl, S. (2018, November). (No) influence of continuous integration on the commit activity in GitHub projects. In *Proceedings of the 4th ACM SIGSOFT International Workshop on Software Analytics* (pp. 1-7). https://doi-org.ezproxy.gsu.edu/10.1145/3278142.3278143
5. Barberis, N., Shleifer, A., & Vishny, R. (1998). A model of investor sentiment. *Journal of Financial Economics*, *49*(3), 307-343.
6. Benner, M.J., Tushman, M.L. (2003). Exploitation, exploration, and process management: the productivity dilemma revisited. *Academia of Management Review* *28*(2), 238-256.
7. Bergquist, M., & Ljungberg, J. (2001). The power of gifts: organizing social relationships in open source communities. *Information Systems Journal*, *11*(4), 305-320.
8. Blanco, G., Pérez-López, R., Fdez-Riverola, F., & Lourenço, A. M. G. (2020). Understanding the social evolution of the Java community in Stack Overflow: A 10-year study of developer interactions. *Future Generation Computer Systems*, *105*, 446–454. https://doi.org/10.1016/j.future.2019.12.021
9. Blau, P.M. (1964). Exchange and power in social life. New York: Wiley.
10. Fong Boh, W., Slaughter, S. A., & Espinosa, J. A. (2007). Learning from experience in software development: A multilevel analysis. *Management Science*, *53*(8), 1315-1331.
11. Bonaccorsi, A., & Rossi, C. (2003). Why open source software can succeed. *Research Policy*, *32*(7), 1243-1258.

12. Chen, I. Y. L. (2007). The factors influencing members' continuance intentions in professional virtual communities — a longitudinal study. *Journal of Information Science, 33*(4), 451–467.
13. Chen, C.-J., & Hung, S.-W. (2010). To give or to receive? Factors influencing members' knowledge sharing and community promotion in professional virtual communities. *Information & Management, 47*(4), 226–236.
14. Chengalur-Smith, S., & Sidorova, A. (2003). Survival of open-source projects: A population ecology perspective. *ICIS 2003 proceedings*, 66.
15. Chou, S.-W., & He, M.-Y. (2011). Understanding OSS development in communities: the perspectives of ideology and knowledge sharing. *Behaviour & Information Technology, 30*(3), 325–337. https://doi.org/10.1080/0144929X.2010.535853
16. Crowston, K., Annabi, H., & Howison, J. (2003). Defining open source software project success. *ICIS 2003 Proceedings*, 28.
17. Daniel, S., Agarwal, R., & Stewart, K. J. (2013). The effects of diversity in global, distributed collectives: A study of open source project success. *Information Systems Research*, *24*(2), 312–333. https://doi.org/10.1287/isre.1120.0435
18. DeLone, W. H., & McLean, E. R. (2003). The DeLone and McLean model of information systems success: A ten-year update. *Journal of Management Information Systems, 19*(4), 9-30.
19. Farjoun, M. (2010). Beyond dualism: Stability and change as a duality. *Academy of Management Review*, *35*(2), 202-225. https://doi-org.ezproxy.gsu.edu/10.5465/AMR.2010.48463331
20. Frank, R. H. (1985). *Choosing the right pond: Human behavior and the quest for status*. Oxford University Press.
21. Fleming, L. (2001). Recombinant uncertainty in technological search. *Management Science*, *47*(1), 117-132.
22. Gelman, A., and Hill, J. *Data Analysis Using Regression and Multilevel/Hierarchical Models.* New York: Cambridge University Press, 2007.
23. Ghapanchi, A. H., & Tavana, M. (2015). A longitudinal study of the impact of open source software project characteristics on positive outcomes. *Information Systems Management, 32*(4), 285-298.

24. Goode, S. (2005). Something for nothing: Management rejection of open source software in Australia's top firms. *Information & Management, 42*, 669–681. https://doi-org.ezproxy.gsu.edu/10.1016/j.im.2004.01.011
25. Hansen, M. T. (2002). Knowledge networks: Explaining effective knowledge sharing in multiunit companies. *Organization Science*, *13*(3), 232-248.
26. Hansen, M. T. (1999). The search-transfer problem: The role of weak ties in sharing knowledge across organization subunits. *Administrative Science Quarterly*, *44*(1), 82-111.
27. Qureshi, I., & Fang, Y. (2011). Socialization in open source software projects: A growth mixture modeling approach. *Organizational Research Methods*, *14*(1), 208-238.
28. Jiang, Q., Lee, Y. C., Davis, J. G., & Zomaya, A. Y. (2018). Diversity, Productivity, and Growth of Open Source Developer Communities. *arXiv preprint arXiv:1809.03725*.
29. Keller, R. T., & Holland, W. E. (1975). Boundary-spanning roles in a research and development organization: An empirical investigation. *Academy of Management Journal, 18*(2), 388–393. https://doi-org.ezproxy.gsu.edu/10.2307/255542
30. Kogut, B., & Zander, U. (1992). Knowledge of the firm, combinative capabilities, and the replication of technology. *Organization Science*, *3*(3), 383-397.
31. March, J. G. (1991). Exploration and exploitation in organizational learning. *Organization Science, 2*(1), 71–87.
32. Morgan, L., & Finnegan, P. (2014). Beyond free software: An exploration of the business value of strategic open source. *Journal of Strategic Information Systems, 23*, 226–238. https://doi-org.ezproxy.gsu.edu/10.1016/j.jsis.2014.07.001
33. Mouakhar, K., & Tellier, A. (2017). How do Open Source software companies respond to institutional pressures? A business model perspective. *Journal of Enterprise Information Management*, *30*(4), 534-554. http://doi.org/10.1108/JEIM-05-2015-0041.
34. Mount, M. P., & Fernandes, K. (2013). Adoption of free and open source software within high-velocity firms. *Behaviour & Information Technology, 32*(3), 231–246. https://doi-org.ezproxy.gsu.edu/10.1080/0144929X.2011.596995

35. Newbert, S. L. (2007). Empirical research on the resource-based view of the firm: An assessment and suggestions for future research. *Strategic Management Journal*, *28*(2), 121-146. http://doi.org/10.1002/smj.573
36. Nuvolari, A. (2005). Open source software development: some historical perspectives. *Eindhoven Centre for Innovation Studies* (available online at http://opensource.mit.edu/papers/nuvolari.pdf).
37. O'Reilly III, C. A., & Tushman, M. L. (2013). Organizational ambidexterity: Past, present, and future. *Academy of Management Perspectives*, *27*(4), 324-338.
38. Podolny, J. M. (1993). A status-based model of market competition. *American Journal of Sociology*, *98*(4), 829-872.
39. Portes, A. (1998). Social capital: Its origins and applications in modern sociology. *Annual Review of Sociology*, *24*(1), 1-24.
40. Raisch, S., & Birkinshaw, J. (2008). Organizational ambidexterity: Antecedents, outcomes, and moderators. *Journal of Management*, *34*(3), 375-409.
41. Rajdeep, G., Gary L., L., & Girish, M. (2006). Location, location, location: How network embeddedness affects project success in open source systems. *Management Science*, *7*, 1043. http://doi.org/10.1287/mnsc.1060.0550.
42. MacLeod, L. (2014, May). Reputation on Stack Exchange: Tag, You're It! In *2014 28th International Conference on Advanced Information Networking And Applications Workshops* (pp. 670-674). IEEE.
43. Sacks, M. (1994). On-the-job learning in the software industry, Westport, CT: Quorum Books.
44. Sampson, R. C. (2007). R&D alliances and firm performance: The impact of technological diversity and alliance organization on innovation. *Academy of Management Journal*, *50*(2), 364-386.
45. Sen, R., Singh, S. S., & Borle, S. (2012). Open source software success: Measures and analysis. *Decision Support Systems,* 52364-372. http://doi.org/10.1016/j.dss.2011.09.003.
46. Seungho Choi, & McNamara, G. (2018). Repeating a familiar pattern in a new way: The effect of exploitation and exploration on knowledge leverage behaviors in technology acquisitions. *Strategic Management Journal, 39*(2), 356–378. https://doi-org.ezproxy.gsu.edu/10.1002/smj.2677

47. Shao, Y., Wu, T., Qiu, H., & Wang, Z. (2018). Ambidextrous activities of internet-based entrepreneurships in Apple App Store: two sides of user feedback. *Technology Analysis & Strategic Management, 30*(10), 1210–1225. https://doi-org.ezproxy.gsu.edu/10.1080/09537325.2018.1458980
48. Deng, S., Huang, Z. J., Sinha, A. P., & Zhao, H. (2018). The interaction between microblog sentiment and stock return: An empirical examination. *MIS Quarterly*, *42*(3), 895-918. https://doi.org/10.25300/MISQ/2018/14268
49. Singh, P. V. (2010). The small-world effect: The influence of macro-level properties of developer collaboration networks on open-source project success. *ACM Transactions on Software Engineering and Methodology (TOSEM), 20*(2), 1-27.
50. Singh, P. V., Tan, Y., & Mookerjee, V. (2011). Network effects: The influence of structural capital on open source project success. *Mis Quarterly*, 813-829.
51. Stewart, K. J., Ammeter, A. P., & 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.
52. Stewart, K. J., & Gosain, S. (2006). The impact of ideology on effectiveness in open source software development teams. *Mis Quarterly*, 291-314.
53. Subramaniam, C., Sen, R., & Nelson, M. L. (2009). Determinants of open source software project success: A longitudinal study. *Decision Support Systems*, 46576-585. http://doi.org/10.1016/j.dss.2008.10.005.
54. Temizkan, O., & Kumar, R. L. (2015). Exploitation and exploration networks in open source software development: An artifact-level analysis. *Journal of Management Information Systems*, *32*(1), 116-150. http://doi.org/11080/07421222.2015.1029382.
55. Robinson, W. N., Deng, T., & Qi, Z. (2016, January). Developer behavior and sentiment from data mining open source repositories. In *2016 49th Hawaii International Conference on System Sciences (HICSS)* (pp. 3729-3738). IEEE.
56. Wang, C. C., & Yang, Y. J. (2007). Personality and intention to share knowledge: An empirical study of scientists in an R&D laboratory. *Social Behavior and Personality: An International Journal*, *35*(10), 1427-1436.
57. Wasserman, S., & Faust, K. (1994). *Social network analysis: Methods and applications* (Vol. 8). Cambridge University Press.

58. Wen Wen, Forman, C., & Graham, S. J. H. (2013). The impact of intellectual property rights enforcement on open source software project success. *Information Systems Research*, *24*(4), 1131–1146. https://doi.org/10.1287/isre.2013.0479

VITA

JOHNSON RAJAKUMAR

BACKGROUND

Johnson Rajakumar is a senior IT executive in the fintech and communications sector. He is a seasoned Information Technology Executive with 20+ years of experience in Product Development, Payment Processing, Architecture, Mergers and Acquisitions, System Integration, Software engineering, Cloud Management and Enterprise infrastructure management in a 24/7 Global Environment. He is consistently recognized as a change agent and an evangelist for agile practices in the financial industry that deliver strong ROI and reduce TCO.

EDUCATION

Doctor of Business Administration, J. Mack Robinson College of Business, Georgia State University, Atlanta, GA. Major Field: Business **Chair:** Dr. Yusen Xia, May 2020

Executive Master of Business Administration, University of Nebraska, Omaha. Major Field: Management, May 2011.

Bachelor of Science, with Distinction, Bellevue University, Nebraska. Major Field: Computer Information Systems, May 2007.

Bachelor of Engineering, with Distinction, College of Engineering, Anna University, Guindy, Chennai, India. Major Field: Electrical and Electronics Engineering, May 1998.

Certifications

Harvard Bok Higher Education Teaching Certificate, Harvard Bok, Nov 2019

Research interests : Information Systems, Deep Learning, Information Security, Decision Making, Open source software, Developer communities, Leadership Style, Quantitative Research, Corporate Restructuring.