The influence of project size on project performance

Through social network characteristics in the Open Source Software community

Hans Biezenaar & Jorrit Jorritsma

Department of Sociology, faculty of Social Sciences

Utrecht University, the Netherlands

Abstract

This study tried to explain the performance of software projects within the OSS-community by looking at the project size and their social network characteristics. Three ways of how project size could affect project performance have been investigated: (1) the direct influence of project size on project performance, (2) the influence of project size on project performance, via social network characteristics, (3) the moderating effect of project size on the relationships between social network characteristics and project performance. Data has been gathered from the SourceForge database, which consisted of more than 30,000 projects. Classical social network theories as well as logical knowledge and mathematical models are used to formulate the hypotheses. Findings show that project size does not affect project performance directly, but rather affects project performance indirectly, going through social network characteristics. Also, project size shows to moderate the relationships between several social network characteristics and project performance.

Preface

Just three years ago, we started our bachelor Sociology at Utrecht University, in which we instantly became friends. We have worked together on multiple projects, as well as in the course ‘Social networks in theory and empirical research’, for which we were both equally motivated. When choosing a topic for our bachelor thesis, the choice for a project on social networks was therefore easily made. We knew beforehand that working with this dataset was going to be challenging, but we have not regretted it any point. Despite the hard times, the work was always fun, very educational and pushed us to limits we have never experienced before. We would specifically like to thank our supervisor David Macro, who offered great support and guidance throughout the entire process. Making this thesis was a great experience. We hope you enjoy reading it as much as we enjoyed making it.

Table of content

1. Introduction

Open Source Software (OSS) projects have been around for several years and are increasing in popularity and prominence ever since, with the biggest platform hosting over 430.000 projects (Daniel & Diamant, 2008; Steinmacher et al., 2016; Sourceforge, 2016). These platforms enable software developers to incorporate third party source code into their work, allowing users to fully exploit publicly available software, thereby assisting in the spread of innovation (Kapitsaki, Tselikas & Foukarakis, 2015). Accordingly, OSS projects are utilized in software development all over the world and are increasingly used by companies in the recent years (Homscheidt & Schaarschmidt, 2016). According to Evers (2000) an Open Source Software project can be described as "*any group of people developing software and providing their results to the public under an Open Source license*".

OSS developers work contemporaneously across a great deal of projects and by doing so, create a network of direct and indirect relationships between them in which they function as both an originator and a receiver of capital. In this, OSS-developers create a network of relations between both developers and projects. These relationships act as channels for the flow of resources in a network, thereby influencing their potential for knowledge diffusion and creation (Coleman, 1988; Granovetter, 1973; Singh, 2010). Analyzing these networks provides a new and interesting opportunity for investigating the influence of network characteristics on project performance. Drawing on the tradition of structural individualism, this study tries to describe the social network structure of OSS-projects by looking at the relationships between both developers and projects.

Previous research has shown that the size of a group influences the performance of that group (Lind & Culler, 2013; Sauer et al., 2007; Emam & Koru, 2008; Liu et al., 2011; Martin et al., 2007). However, relatively little research has been conducted on how project size influences project performance within the information technology (IT) area specifically, as most research on this topic focuses on R&D teams and industrial units (Lind & Culler, 2013). An even smaller amount of studies has applied a social network approach in explaining the effect of project size on OS performance (Van den Broek & Westra, 2015). Although there have been some studies which tried to explain OSS project performance by social network characteristics (Singh, Tan & Mookerjee, 2008; Grewal, Lilien & Mallapragada, 2006), no study investigated what role project size could play in this mechanisms. Furthermore, there is still no consensus within the IT area on the direction and nature of the relationship between project size and project performance, nor clarity about the underlying mechanisms that cause them.

This research focuses on the direct relationship between project size and project performance, as well as its indirect influence on performance through certain social network characteristics of projects. We specifically want to investigate how project characteristics influence project performance and how network characteristics such as brokerage and closure moderate this relationship. Socially, it is relevant to increase understanding of how network characteristics influence one another, as it helps to understand how network structures and network behavior operates on a fundamental level. We specifically chose to focus our research on factors such as project size and project performance, as these are concrete concepts that are used in everyday (professional) life. Findings about these concepts can thereby be implemented in all layers of society in which project-based working is used. Furthermore, the size of a group is a variable which differs considerably between different projects and influences a great deal of network characteristics, making for a more diversified and in-depth analyses. Lastly, analyzing relationships between these network characteristics within the OSS community makes for very compelling and reliable research, as it provides us with a large dataset of 32 000 respondents and contributes knowledge to a naturally innovative area (IT).

Our study contributes to the existing literature by offering an in-depth analysis on the relationship between project size and project performance and how this relationship can be explained by social network characteristics like brokerage and closure. By doing so, this research fills the gap in literature about the relation of project size and social network characteristics of projects. Furthermore, we will go into detail on the moderations that occur between brokerage, closure and project size and how these influence the relationship between project size and project performance, which is something that has never been done before. By looking at the possible interactions between brokerage, closure and project size, this study looks further than the main effects of these factors and tries to find possible interactions that have never been investigated. These variables and relationships will be further discussed in the theory section.

The research question we aim to answer is: *"How does project size influence project performance via social network characteristics and how does it moderate the relationship between social network characteristics and project performance?"*

2. Theory & Rehypotheses

2.1 Introduction

In this section, the theory and derived hypotheses will be discussed. First, a short paragraph will be devoted to the explanation on indicators of project performance, after which we will deal with the theory and hypotheses about project performance. We consider two factors to have an impact on project performance: project size and the level of brokerage of a project. After explaining these relationships, we will go in depth about the theory and hypotheses concerning the relationship between project size, brokerage and closure. Lastly, a paragraph will be spent on several moderated relationships between brokerage, closure and project size on project performance.

2.2 Predictors for project performance

2.2.1 Project size as a predictor for project performance

Previous research has shown the size of a group to influence the performance of the group (Lind & Culler, 2013; Sauer et al., 2007; Emam & Koru, 2008; Liu et al., 2011; Martin et al., 2007). However, little research has been done on this relationship, due to the complex nature of the relationship. Project size is often used as a synonym for project complexity, as information technology projects need more people as their complexity increases (Lind & Culler, 2013). Most research on this topic focuses on industrial units and R&D teams. The theories and results found in this research do not fully apply to OSS-projects, as OSS-projects do not share specific properties with most other teams or groups. OSS-projects consist of developers who mostly do not know each other in real life, which makes the ties between them rather loose. Furthermore, an OSS-project consists mostly of members who participate on a voluntary basis. This means that reprimands for being unproductive will be unlikely to take place in these OSS-projects. The lack in formal institutions within OSS-projects is, however, compensated by the sheer amount of informal institutions. These special characteristics need to be taken into account when constructing the hypotheses.

Results from the studies on the effect of group size on group performance vary significantly. In 1966, Wallmark and Sellerberg published an article about the effect of team size on the efficiency within research teams. They found an increase of efficiency within a team when a team has more team members. These findings were confirmed a few years later in a study of Wallmark, Holmqvist, Eckerstein and Langered (1973). The authors of this study mentioned two possible explanations for their findings: team members in larger teams have accessibility to better equipment and team members of larger teams have more choice to select the people with whom they want to cooperate with. Furthermore, a study by Dailey (1978), found that team size has a positive effect on the process within a team (i.e. problem solving) as well on the productivity of a team.

In contrary to former literature, Alan and Ingham (1974) concluded that a growth in group size has a negative impact on the performance of a group. Their study focuses on the Ringelmann-effect, which is the decrease of performance as the group size increases, due to a lack of motivation among the group members. This phenomenon is also related to the concept of “free-riding” or “volunteer dillema”, which is the tendency for a person within a team to not fully commit to the common good within a team, as the low commitment of that one person goes unnoticed in a group of large size (Isaac, Walker and Thomas, 1984; Archetti, 2009). Besides the lack of motivation, the difficulty to coordinate within a large group can also be seen as a reason for why the performance drops as the group gets larger.

Within the information technology area, there is still no consensus on the direction and nature of the relationship nor clarity about the underlying mechanisms that cause them. Most studies, however, lean towards a negative effect of project size on project performance, suggesting that smaller projects perform better than bigger ones. (Lind & Culler, 2013; Sauer et al., 2007; Emam & Koru, 2008) Research from Sauer et al. (2007) opened the way to more research on this topic, as they argued that the relationship between project size and project performance is complex and is not of a simple linear nature. They found that smaller projects generally perform better than bigger ones, but also found that some larger projects outperformed smaller projects, suggesting further research on large projects was necessary. In their exploratory study about the critical success factors of IT project performance, Lind & Culler (2013) found a negative significant relationship between project size and project performance. A similar result was found in the evaluative study by Emam & Koru (2008). They found that smaller projects tend to have lower cancellation rates, offer better quality, are on budget more often and are on schedule more often, causing them to be more successful than larger projects.

Previous research has shown that project size significantly influences project performance. The majority of both research on information technology projects and R&D, show evidence for a negative relationship. Furthermore, research in the information technology area has shown this relationship to be non-linear. Lastly, research such as that from Sauer et al. (2007), show how size can have very incompatible influences on performance and raise questions as to how the relationship is shaped. A project with very little developers will most likely never be able to reach the same performance as a project with more people working on it. Conversely, when a project gets too large, the coordination between developers will decrease and will lead to weak performance. We expect that, in order for a project to reach its full potential in terms of performance, there needs to be an optimal number of developers working on a project. We therefore hypothesize that:

H1: The relationship between project size and project performance is inverse-u-shaped.

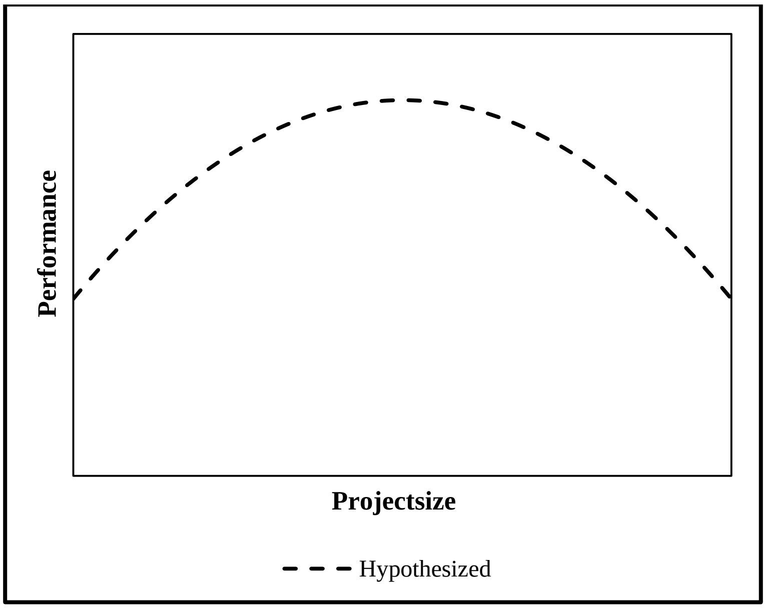

Figure 1. The inverse-u-shaped relationship between project size and project performance.

2.2.2 Brokerage as a predictor for project performance

According to the structural hole argument, social capital is the information and control advantages of being the broker in relations between people who would otherwise be disconnected from one another, where the disconnected people stand on opposite sides of a structural hole (Burt, 1997). Because of this, a structural hole is an opportunity for the broker to take advantage of the information between people and control which projects and people from opposite sides of the hole come together. This argument comes from a line of network theories in Sociology during the 1970s by Granovetter (1973), Freeman (1977), Cook & Emerson (1978) and Burt (1980).

This advantage in information and control within a network has a lot of positive influences on the performance of that network. Being the broker in a network, provides access to information far beyond what could be attained without that position, as it makes that information more reliable, unique and is received much faster. Furthermore, a broker has a say in whose interests are served by the bridge. The disconnected contacts are very likely to communicate through the broker, giving the broker a complete and accurate picture of the situation (Burt, 1997).

A large part of social network research on the relationship between brokerage and performance, has shown there to be a positive association between the two. (Burt, 2005; Hansen,Podolny and Pfeffer, 2001; Krackhard & Stern, 1988). In his research on social capital, Burt showed that managers with discussion partners in diverse groups had a high chance of being promoted or to receive an above-average raise, whereas managers who only had a small isolated group of discussion partners who only discussed with each other, were much less likely to be promoted. (Burt, 2005) They also showed to have more positive evaluations and received higher compensations for their work.

Team based performance has shown to be positively correlated with high levels of brokerage by quite a number of studies. In their research on organization network structures, Hansen, Podolny and Pfeffer (2001) found that teams reached completion more quickly when they were in frequent and close contact to other divisions, showing that the most successful teams are the teams who have intergroup communication. Other research on students found that higher group performance was positively related to cross-group friendships (Krackhard & Stern, 1988). Stuart & Podolny (1999) showed that firms who engage into contact with firms outside of their own technological area, are more likely to innovate than firms who only stay in their own sector.

In a network such as the OSS-projects - where contact between projects is uncommon- it is expected that the number of structural holes is high. This should give an advantage to those projects who are able to build intergroup bridges that span those structural holes, as they are able to gain more unique and high quality information faster than other projects. This, accompanied by results from previous research on the effects of brokerage on project performance, makes for the second hypothesis:

H2: Brokerage is a positive predictor for project outcome performance.

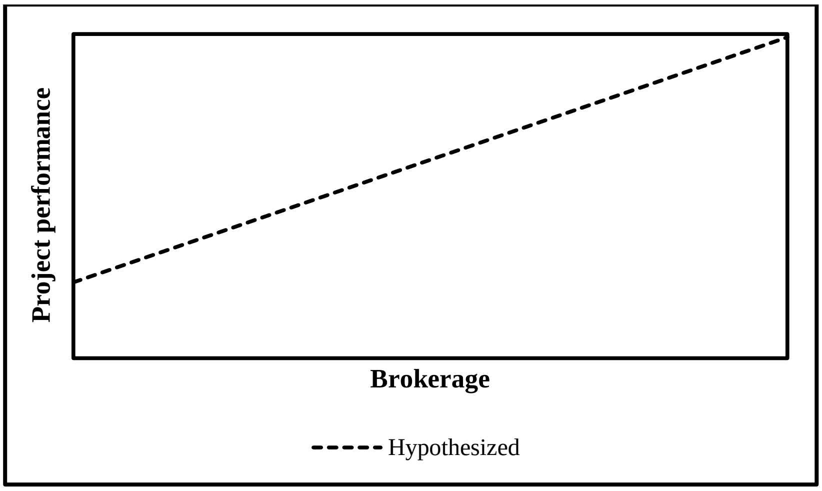

Figure 2. The positive relationship between brokerage and project performance.

2.3 The relationship between project size and brokerage and between project size and closure

2.3.1 Introduction

This section will be dealing with the relationships between project size and brokerage and project size and closure. First, the relationship between project size and brokerage will be discussed. This will be done by a 2-staged hypothesis: we will first explain why project size affects the number of ties that a project acquires, and then why the number of acquired ties on its turn influences the brokerage of the project. After that, we will come to the relationship between project size and closure. Again, we do this in two stages. First we predict the influence of project size on the number of ties and predict afterwards the relationship between the number of ties and closure.

2.3.2 Project size as a predictor for brokerage

Brokerage has been shown to influence project performance (Burt, 1997; Burt, 2005; Hansen, Podolny & Pfeffer, 2001; Krackhard & Stern, 1988; Stuart & Podolny, 1999). In turn, the level of brokerage is affected by certain properties of the project.

First of all, we expect project size to have an impact on the brokerage of a project. As mentioned earlier, brokerage can be measured in many ways. This study uses the centrality measure *level of betweenness* to measure the brokerage of a project, thus betweenness and brokerage will be used interchangeably throughout this study. We will go in more depth on these terms and their use within this research in the data & methods section. Very little research has been done on the relationship between project size and the level of brokerage. For that reason, the following theories will, for the main part, be based on logic and general knowledge about social network behavior.

To predict the relationship between project size and the level of brokerage of the project, we first need look at the micro-level of the OSS-community. A tie between two projects is constituted when the two projects share a developer. In social network theory, the number of ties is often referred to as the "actor's degree", whereby a high level of degree corresponds to a high number of ties (Freeman, 1978). Every developer within a project is likely to create several number of ties to other projects, due to the fact that the developer is working for more than one project at a time. The more developers a project has, the more developers constitute ties to other projects, that is to say a project with more developers will naturally acquire more ties to other projects. However, an increase in the number of developers within a project also increases the likelihood that ties with other projects are going to be shared within that project. This means that, as the number of developers within a project increases, the increase of degree caused by project size will decline. Therefore the third hypothesis will be:

H3: Project size is a decreasing positive predictor for the number of ties that a project has.

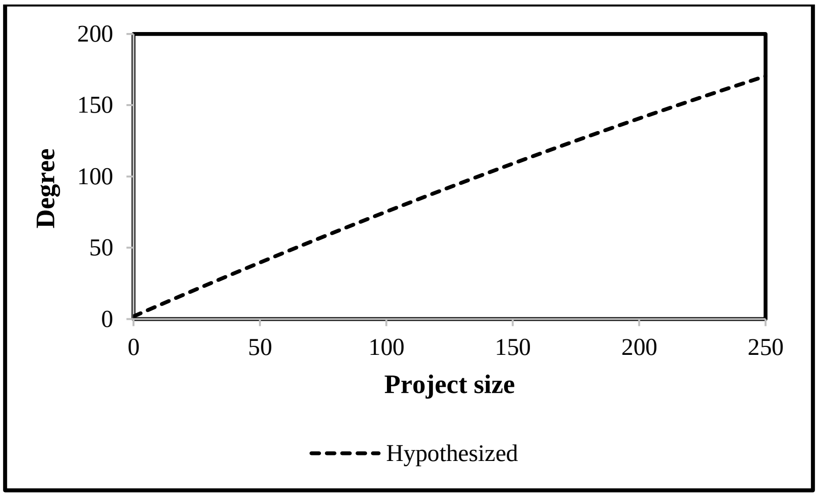

Figure 3. The decreasing positive relation between Project size and degree.

It is likely that the degree of a project affects the level of brokerage of the project. Research about the connection between *degree* and *betweenness* is very scarce, but some attempts have been made in the past. A study on the correlations between network centrality measures by Rothenberg et al. (2004), showed that *degree* and *betweenness* were highly positively correlated to each other. Another study, investigating the degree/betweenness distribution of airports, found the same positive relation (Guimerà & Amaral, 2004). A few years later, Valente et al. (2008) conducted a study about the correlations between eight centrality measures, including *degree* and *betweenness*. This study found the correlation between *degree* and *betweenness* to be the second highest correlated centrality measures.

Based on the findings of earlier research, we expect to find at least a positive correlation between degree and betweenness. However, it is still unknown whether this positive relationship between degree and betweenness is linear or quadratic. In our opinion, logic would strongly suggest the latter to be true, as a higher degree does not only imply more direct ties, but also more indirect ties. This higher number of indirect ties can have a huge impact on the *betweenness,* since betweenness is measured by the number of nodes which have to go through a single actor (ego) to reach a certain other node (alter) (Everett & Borgatti, 2005). The more indirect ties the project has, the more indirect ties might need to go through the project to reach other projects, thus increasing the level of brokerage

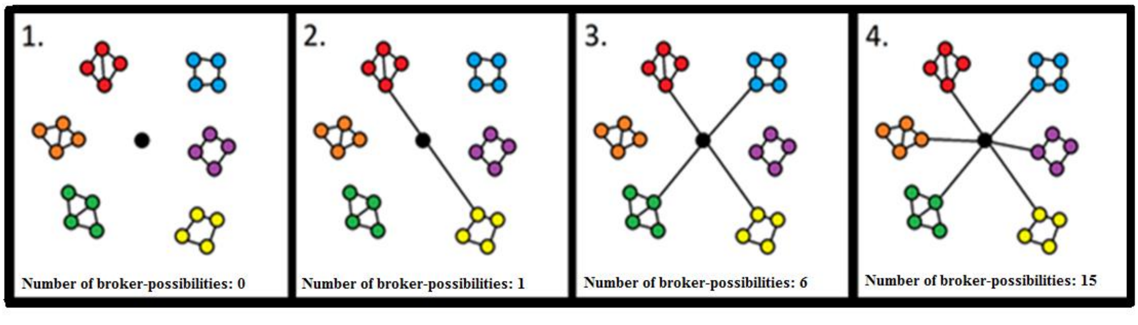
1.
Number of broker-possibilities: 0
2.
Number of broker-possibilities: 1
3.
Number of broker-possibilities: 6
4.
Number of broker-possibilities: 15

Figure 4. An example of the relationship between degree and betweenness of a project

Figure 4. shows a more specific image of how *degree* can affect the level of brokerage. The first picture shows the OSS-network community with the black node as the project. This OSS-community consists of six clusters of other projects, which is illustrated by the different colors. In the second picture, the project has acquired two new ties, one with a red node and one with a yellow node. This gives the project the opportunity to broker between the red and yellow cluster. This leads to the occurrence of *one* broker possibility, namely between the red and yellow cluster. In the third picture, the project again has two more ties: one with a node in the green cluster and one with a node in the blue cluster. This increase of two ties immediately results in a substantially higher betweenness for the project, as the project functions not only as a broker between the red and yellow cluster, but also between the red and blue, red and green, blue and yellow, blue and green and yellow and green cluster. The number of clusters between which ego can broker grows exponentially, while only two new ties have been constituted. The two new ties lead to an increase of five new broker possibilities for the project, which brings the total on six broker possibilities. In the fourth picture, this is again illustrated: by adding two more ties, the number of clusters between which ego is able to broker, increases exponentially to fifteen.

An important note here is that this logic only applies when brokerage is measured with *betweennes* centrality. Another common measurement for brokerage is *constraint*, which is the extent to which a person's network is concentrated in redundant contacts (Burt, 1997). Constraint is high in a dense network, meaning that everyone is directly connected to one another, or in a hierarchical network, meaning that everyone is indirectly connected via a central contact.

Constrained networks span fewer structural holes, which, according to the structural hole argument, means a lower level of brokerage (Burt, 1997). Brokerage can thereby be measured by taking 1-constraint. If brokerage is measured as 1-contraint, then even if constraint is low, it could be possible that the betweenness of a network is still high, because the project is situated in a closed network but still has one tie that connects different networks network. For that reason brokerage - measured as 1-constraint – would not be suited for this hypothesis. But, since we use betweenness as a measurement for brokerage, the hypothesis holds.

The increasing positive relationship between *degree* and *betweenness* was also found by Macro (2016). In this study, Macro performed several simulations on the relationships between degree and betweenness for different degree distributions and found the relationship to quadratic. Results from these simulations are displayed in figure 5. Because our own data was found to have a degree distribution of 1,5, which is normal for social networks (Madey, Freeh & Tynan, 2002), we pay attention to the line which applies to a degree distribution of 1,5 in the graph (the orange line). The orange line shows that the increasing positive relationship between *degree* and *betweenness* is the strongest for a degree distribution of 1,5, which indeed corresponds with our own data.

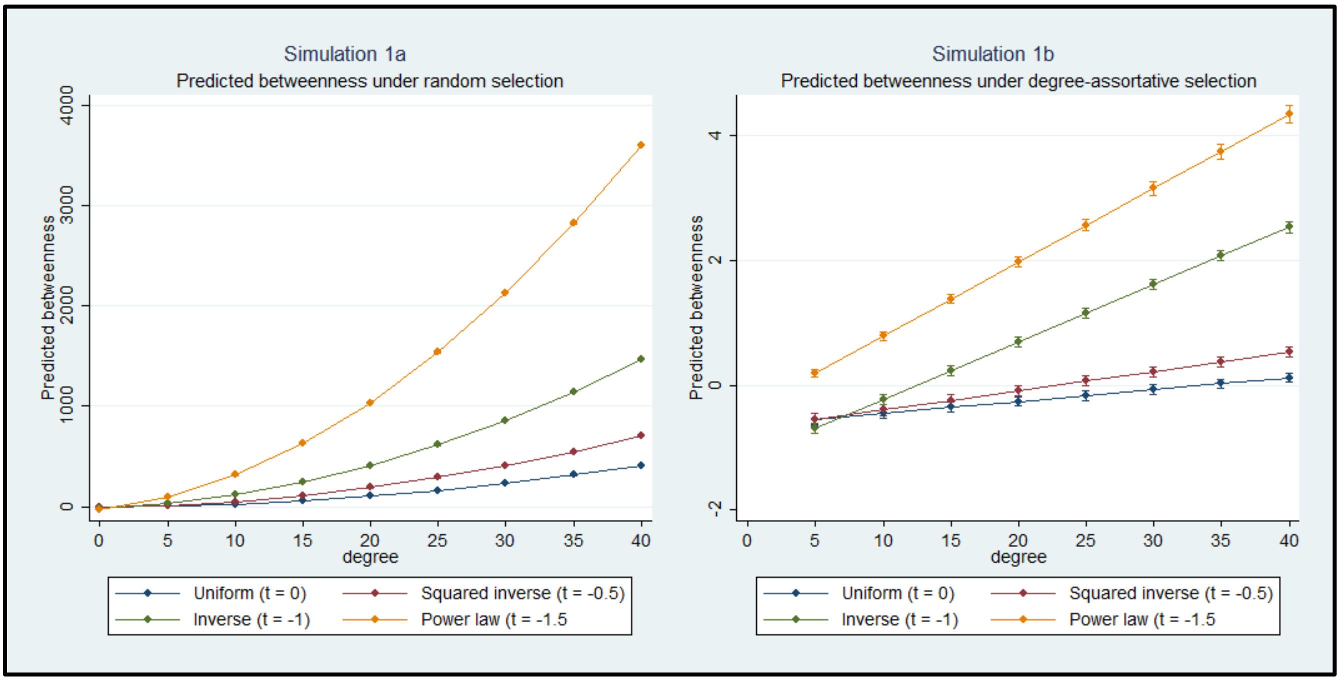

Figure 5. The increasing positive relationship between degree and betweenness found by Macro (2016).

Based on the aforementioned theoretical arguments and the findings of Macro (2016), we would expect an increasing positive relationship between *degree* on *betweenneess*. Hypothesis 4 will for that reason be:

H4: The relationship between degree and brokerage is increasingly positive.

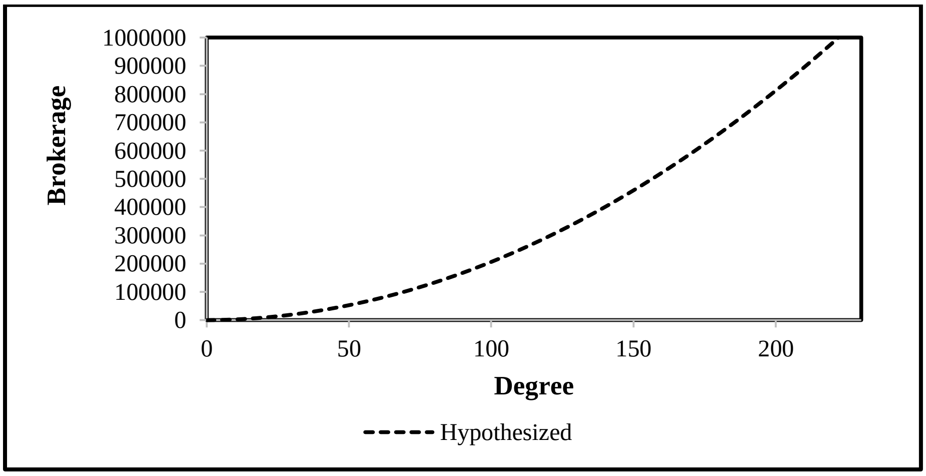

Figure 6. The increasing positive relationship between degree and brokerage.

2.3.3 Project size as a predictor for closure

When there are more developers within a project, it naturally results in a higher chance for an increase higher level of degree, since most developer will create ties to other projects. Little research has been done on the effect of degree on closure and the research which has been done is not unanimous about whether this effect is positive or negative. For example, in their study about community size and closure, Alcott et al. (2007) found a higher closure for students who have a high level of degree (measured by the number of friends) than for students with a low level of degree. This would imply that a higher degree leads to higher closure. Unfortunately, the study lacks comprehensive explanation for this conclusion. Also, it would be unwise to distract our hypothesis directly from the found result in the study of Alcott et al., because their study focuses on the social networks of students, causing generalization issues for their results. It is therefore not unlikely that other mechanisms are in place for the networks of OSS-projects.

In contrast to the study of Alcott et al, a study by Fowler et al. (2009) showed a likelihood for a negative effect of degree on closure. The researchers came to this conclusion by performing several simulations based on fictional datasets. Further research about the possible effect of degree on closure is missing. For that reason, it is hard to derive any hypotheses based on former literature.

Based on knowledge of social network behavior, one would assume the relationship between project size and closure to be negative rather than positive. Closure can be defined as the relative number of ties of the project which are interconnected to each other. Closure is therefore calculated by counting the number of ties of the project that are connected to each other, divided by the total possible connections between the ties. When a project has few ties, for example two or three, the chance exists that two of the ties are connected to each other. When the degree of a project increases, the number of possible connections between the ties of the project increases exponentially as well. However, the ties of the project are only able to make a limited number of ties, which means the actual number of ties between them most likely grow gradually rather than exponentially.

One simple method to calculate the level of closure within a project, is to divide the number of ties of the ego within the project by the maximum number of possible ties between ties of the ego project. This would give the following formula (in which AA means the number of ties between the alters):

$$Closure = \frac{\sum_{obs} AA}{\sum_{pos} AA}$$

The number of *possible* ties is dependent of the number of alters that a project has. The total of possible ties between the alters of the project can be calculated as follows:

$$\sum_{pos} AA = \frac{(N_{alters} - 1) * \ N_{alters}}{2}$$

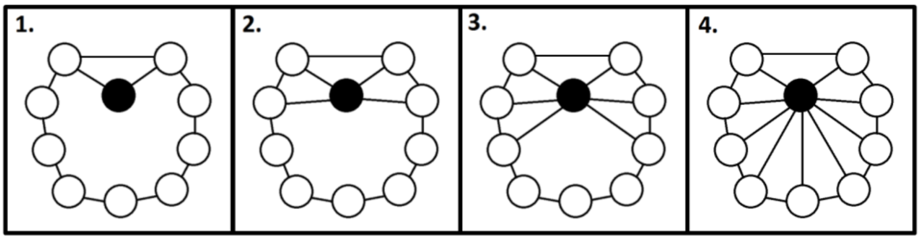

Figure 7. An example of the relationship between degree and closure

Figure 7. shows how degree affects closure. It is assumed that every project is able to maintain two ties to other projects, because the developers who create the ties are restricted by time. Only the focal-project is able to acquire more than two ties because it has so many developers. This is depicted in the first picture. In the second picture the ego project acquires two more ties. Based on the theory of Granovetter (1972) about triadic closure, we expect that the two new ties have a common tie with the ego project. What happens is the actual number of ties between ties of the project increases from one to three. However the number of *possible* ties between ties of the project increases from two to six. In the third picture the ego project acquires again two more ties, which brings the number of actual ties between ties of the project from three to five. However, the number of *possible* ties between the ties of the project increases exponentially from six to fifteen. In the last picture, when the ego project has reached a maximum degree, the actual number of ties between ties of the project has become ten and the number of possible ties between the ties of ego project has grown to forty five.

An increase in degree should mean that the constantly growing actual number of ties is divided by an exponentially growing number of possible ties between ties of the project. In a mathematically sense, this implies that an increase in degree is negatively related to closure, but this negative relationship will lessen when degree increases. This is the decreasing negative relationship between degree and closure. Based on the theory above, we hypothesize that:

H5: Degree is a decreasing negative predictor for closure.

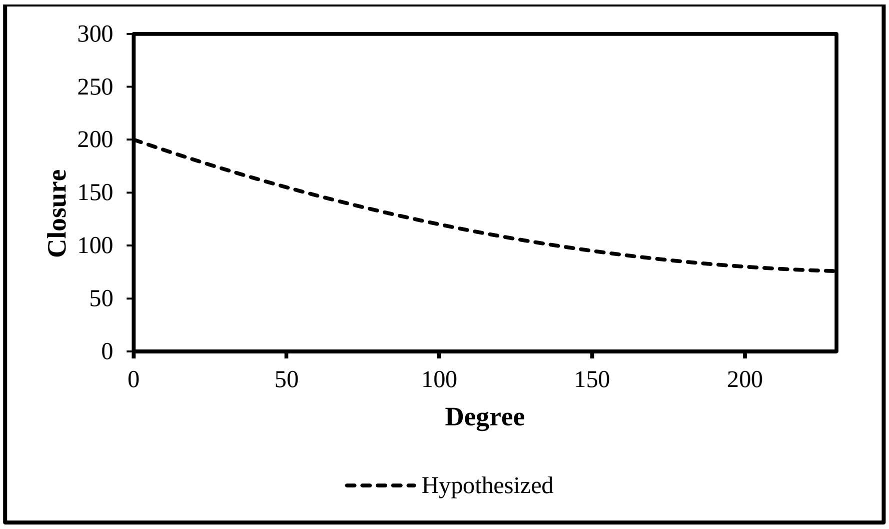

Figure 8. The decreasing negative relationship between *degree* and *closure*

2.4 Moderations of closure and project size on the relationship between brokerage and project performance

2.4.1 Closure as a moderator for the relationship between brokerage and project performance

On the surface, one would say brokerage is negatively related to closure, as they seem like near opposites of one another. After all, project size is positively related to brokerage and negatively related to closure. However, Burt (2000) showed that a synthesis of both is possible. Figure 9 illustrates an ego's network which has both high brokerage and high closure. A substantial part of the nodes in this network are connected to one another, yet the level of brokerage is relatively high as well.

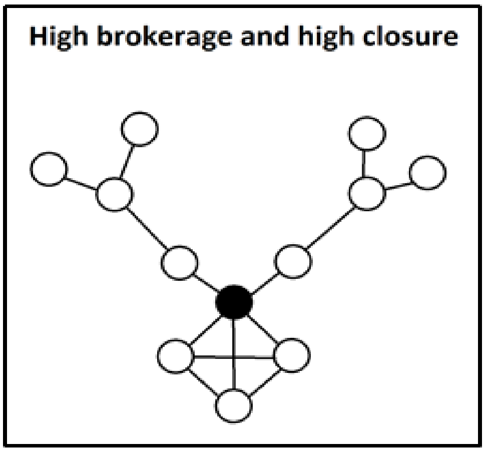

Figure 9. An example of a network with high brokerage and high closure (Burt, 2000).

Burt is commonly known for his literature on social capital and brokerage, which strongly emphasizes the positive role of brokerage on performance. However, even Burt (2000) agreed that closure can play a positive role in the performance, as he suggests that the positive relationship between brokerage and performance is dependent on the closure of the ego's network. Burt reasons that a network with a very high level of brokerage will not fully benefit from that brokerage, when there is a total lack of closure within that network. This is the problem of disorganization: when a project has a network with high brokerage but no closure at all, it jeopardizes the cohesion for the ego. In our study, a project with high brokerage, implies that many developers within the project are *also* working in other projects. It is likely that this causes problems with coordination which eventually lead to disorganization.

Burt proposes that closure can mitigate this problem of brokerage. Closure can create certainty, trust and improvement of communication (Burt, 2000). For the OSS-projects with a network with high brokerage, a small amount of closure could be beneficial for both brokerage and performance. The danger of disorganization will thereby be lessened when the network of the project has a small amount of closure, as this will keep the developers aware of each other's project and improve their communication.

We assume that there is a certain degree of closure which is optimal, but also allows for the possibility to acquire new information by structural holes. The moderation effect of closure on the positive relationship between brokerage and project performance will therefore be from low, to high, to low, that is to say the moderation effect of *closure* on the positive relationship between *brokerage* and *project performance* would be inverted u-shaped. The following hypothesis has been derived from this:

H6: Closure moderates the positive relationship between brokerage and project size positively, until an optimum, then negatively

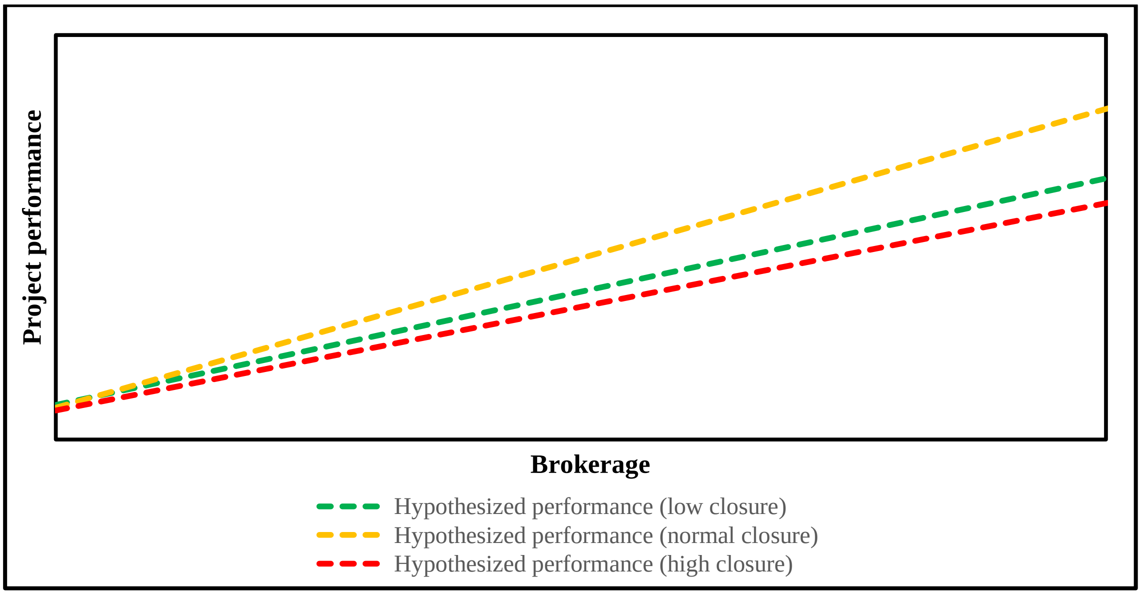

Figure 10. The moderation effect of closure on the positive relationship between brokerage and project performance.

2.4.2 The influence of project size on the moderation of closure on the relationship between brokerage and project performance

A certain level of closure is necessary to fully benefit from the positive effect of brokerage on project performance. As discussed above, this is due the fact that zero closure will create the risk for a project to be literally “forgotten” by its developers. A certain level of closure can keep the developers aware of each other’s project. It is arguable that smaller projects need this certain level of closure more than large projects, since the small projects have the biggest risk to be “forgotten” by its own developers, simply because there are more stakeholders.

For that reason, we expect the moderation of closure on relationship between brokerage and project performance -which was predicted in last paragraph- to be weaker for large projects. We hypothesize that:

H7: Project size negatively affects the moderation effect of closure on the positive relationship between brokerage and project performance.

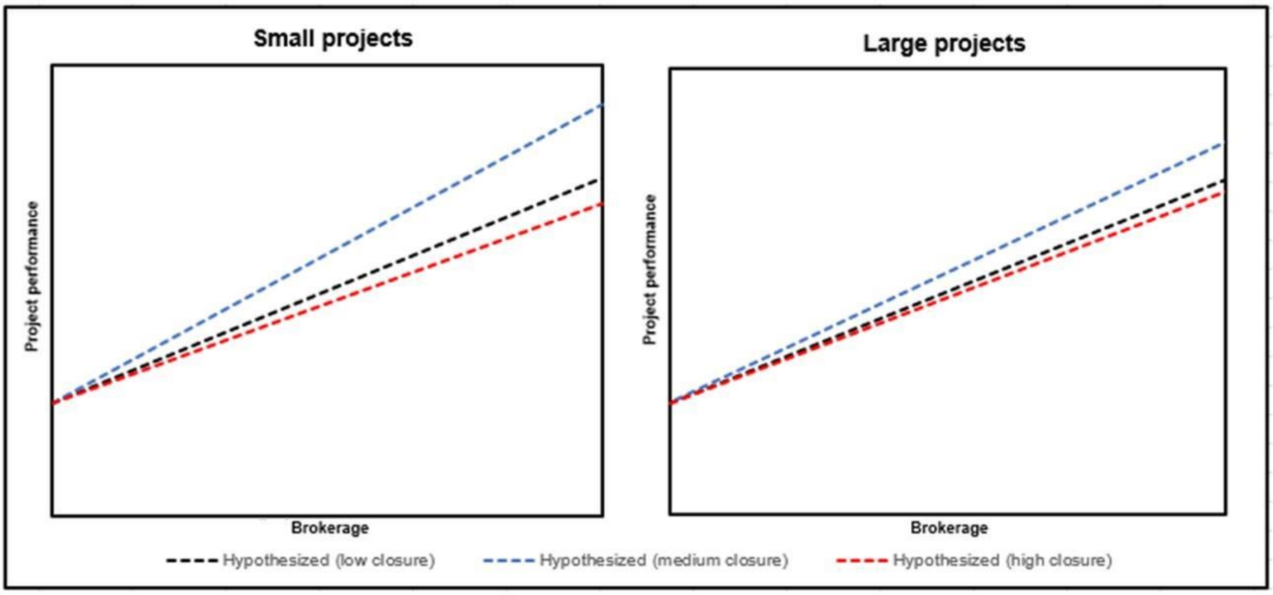

Figure 11. The moderation effect of closure on the relationship between brokerage and project performance, separated by project size.

2.4.3 The influence of project size on the relationship between brokerage and project performance

Project size could also have a mitigating impact on the positive relationship between brokerage and project performance. As stated above, brokerage can help projects to gain new relevant information to perform better. Especially for small projects with few developers it can be really useful to be in a broker position. Conversely, larger projects with many developers mostly have differentiated the tasks within the project. Hence, larger projects already have all relevant information in their project, because they have so many developers. For a larger project it is therefore less attractive to be in a broker position than it is for a small project. Therefore, we expect that:

H8: The positive relationship between brokerage and project performance, is negatively moderated by project size.

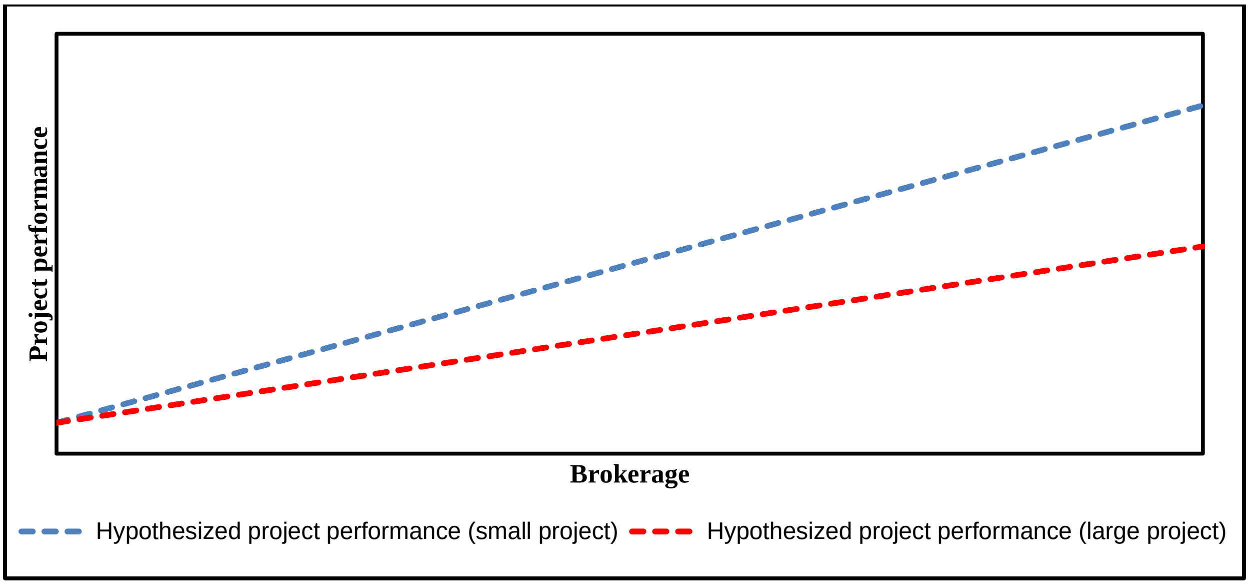

Figure 12. The negative moderation effect of project size on the relationship between brokerage and project performance.

3. Data & methods

3.1 Introduction

In the following section, attention will be paid to the data and used methods for our analysis. First, information will be given about the data which is used in this research. Then we discuss the operationalization of the independent and dependent variables. Finally, we explain which control variables have been chosen and how they were operationalized.

3.2 Data collection and sampling method

Data for our analysis is gathered through an OSS community resource called *SourceForge*. *SourceForge* is a web-based service for software developers which serves as a centralized online location to control and manage open-source software (OSS) projects on. It has over 430.000 projects and hosts more than 3.7 million registered users (SourceForge, 2016). Due to its popularity and large amount of developers and projects registered on the website, *SourceForge* can be considered the most representative of the OSS movement (Singh, 2010). The SourceForge dataset was obtained on a website called *FlossMole,* which is an organization that collects OSS projects so it can be used for scientific research (Howison, Conklin & Crowston, 2006).

The network characteristics of the OSS-projects are abstracted from the raw data by approaching the data from a structural individualistic point of view. Assumed is that two projects are connected to one another if they have a developer in common. This means that if a developer works for two or more projects, these projects are connected to each other. This is illustrated in figure 13. By determining the ties for every projects, it was possible to construct a total social network overview of all the projects within the community of *SourceForge*. After this, the necessary social network characteristics for every project were derived, such as degree, brokerage and closure.

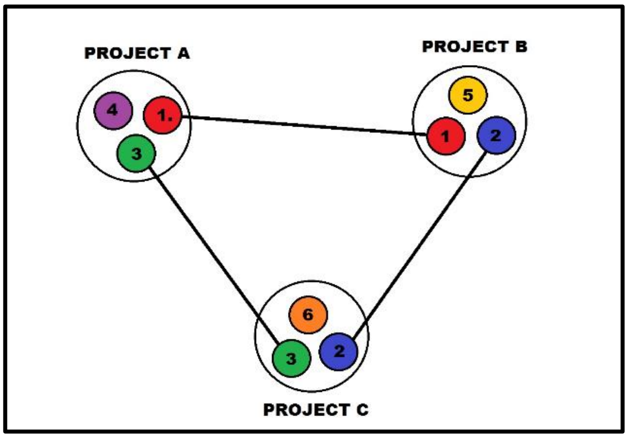

Figure 13. Projects are tied to one another when they have a developer in common.

The dataset contains N=30,031 active projects. Only active projects were taken into consideration, as inactive projects perform few activities and make no code contribution, making them less suitable for the analysis we want to perform (Westra & Van den Broek, 2015). The dataset comprises a number of project characteristics (i.e. the age of the project, the number of developers on a project), indicators of project performance (i.e. the number of downloads, the number of website hits) and several network-measures (i.e. betweenness centrality, transitivity). Important to note is that these measures are on a project basis rather than a single user-developer basis, where a project can be described as a group of user-developers working on the same software (Westra & Van den Broek, 2015).

3.3 Operationalization of the independent and dependent variables

3.3.1 Size

The explanatory variable used in this study is the size of an OSS project. Size is measured by the number of registered developers that is in a certain project.

3.3.2 Brokerage

As mentioned before, the level of brokerage is operationalized as the level of betweenness centrality. Betweenness is one of the most valid centrality measures to measure brokerage (Freeman, 1977; Burt, 2000). It measures the number of times a node acts as a bridge along the shortest path between two other nodes. It thereby functions as an indicator for brokerage, as it measures the extent to which a person brokers between individuals in a network.

The dataset contains six items on betweenness centrality, where each item has a different cut-off point in path length. The path length is the number of nodes counted from ego to the last node in the chain. This means that if the cut-off length is 3, only nodes who are within 3 nodes of ego are taken into account. The six items range from a cut-off path of three, to a cut-off path of eight. The items on betweenness centrality are all highly correlated, meaning the items all function equally well in our analysis. An argument could be made about the information degradation after certain path lengths, where information becomes unreliable when path lengths are too long. Extremely long path lengths are therefore not relevant, as information does not reach its designated goal. We therefore chose to use the medium of the six items, which roughly comes down to a cut-off path length of 5.

84% of the OS projects have a betweenness centrality of 0, meaning that 84% of the projects do not function as a bridge between other projects. This means that any actor that does not score 0 on this variable automatically has an extremely high score. These factors cause for a highly skewed distribution that needs to be altered to be functional. This is done by computing a logarithm for the variable betweenness, which makes the distribution symmetrical by transforming the values of brokerage to orders of magnitude. By transforming the values to orders of magnitude, the larger values get aggregated to the same order of magnitude. This is how a more normal distribution of values arises. (Log) betweenness will then be used as the indicator for the level of brokerage of OS projects. We used the log+1 to compute the logarithm for betweenness, as it impossible to calculate a logarithm with a betweenness of 0. The simple formula for calculating this is:

$$(log)Brokerage = log(Betweenness5 + 1)$$

3.3.3 Closure

Closure can be defined as the degree in which the ties of ego are interconnected with each other. Closure is operationalized by the transitivity index of a project, where transitivity merely functions as a synonym for closure. This index measures the average fraction of a developer's coworker who also work together with each other: the higher the proportion of transitive relations in a network, the higher the level of closure in a network. Important to note here is that we use a local measurement for closure that reasons from ego. In other words: we look at the proportion of closed triads of which ego is also a part.

3.3.4 Project performance

Indicators of the predictive variable performance are number of downloads, number of forum posts, number of project pages and number of website hits, all measured over a 60 day interval. For our research, we will operationalize performance as the number of times a project gets downloaded. In our view this is the most valid indication for the performance of a project. Previous research has also used the number of page views as their indicator for performance. However, as the items do not strongly correlate with one another (*r* (30029) = .21, *p* < .001), a choice has to be made between either one of them. We therefore chose not to incorporate the number of page views in our research, as we strongly believe the number of downloads to be a far better indicator of project performance. After all, software will be likely to get downloaded more often when it guarantees a certain quality. Also, a number of other studies consider the number of downloads as a valid variable to measure the performance of OSS projects (Crownston, Annabi & Howison).

By measuring the performance of the project by the number of downloads in a 60 day interval, the performance of the project is measured as so called *outcome performance*. Earlier research has made the distinction between process performance and outcome performance, in which process performance refers to the process within the project (Van den Broek & Westra, 2015). One can think about the coordination between the developers within the team as an example for process performance. For our study, we have considered to not include a possible distinction between process performance and outcome performance. The first reason for this is that we expect that process performance will affect outcome performance. When the process within a project is deficient, the outcome performance will be disappointing as well. By measuring the project outcome performance, we thereby take the possible effect of the process within the project into account as well. The second reason for focusing only on the outcome performance is, because of potential practical burdens. When we would distinguish process performance from outcome performance this would lead to almost twice as many hypothesis. This, combined with the fact that a large portion of the process performance is already taken into account, lets us to believe that adding a measurement for process performance in our analyses does not provide sufficient benefits to out way the limitations.

The variable of 'number of downloads in a 60 interval' initially had several problems we had to address before it was usable for our analysis. The first problem was the possible effect of project size on the number of downloads. It is arguable that projects with more developers on them are meant for a bigger market and will therefore always be downloaded more than smaller projects. This problem is solved by making the number of downloads relative to the project size. This was done by dividing the number of downloads in 60 days by the number of developers of the project. After this, the distribution of the variable showed to be highly right skewed. Because our statistical analysis are based on the assumption of a normal distribution, a logarithm has been applied to the variable.

Another possible problem that arose here, was that making the variable relative to the number of developers and taking the logarithm of the variable afterwards, meant that the number of developers are also included in the logarithm. Another possibility would be to make a logarithm of the number of downloads first and then to divide it by the number of developers. We have examined whether our method causes problems for the analysis and did conclude this was not the case. First of all, the number of developers is also rightly skewed, so a logarithmic transformation will be a useful solution for variable as well. Besides this, a correlation was performed between outcome variables for the first and the second method. This showed a high correlation between both methods, r (30029)= .84, p < 0,01. Therefore we could say both methods are interchangeable. As the first method also addressed the issue of a skewed distribution for the number of developers, we opted for this method. This gives the following formula for project performance:

$$(log)Project\ performance = \log \frac{Number\ of\ downloads}{Number\ of\ developers}$$

3.4 Control variables

3.4.1 Development status

The first control variable is the development status of the project. The development status contains the stage in which the project is situated. The variable has five values, which are: 'Developing', 'Alpha', 'Beta', 'Stable' and 'Mature'. Expected is that the development status affects the performance of the project, as a project which is in an earlier phase is less likely to have a high number of downloads than a project which is already in a further stage.

3.4.2 Program language

The software which is created by a project can be coded in several so called program languages. Mostly the code for software is coded in one specific program language, which cannot easily be changed to another program language. The type of program language which is used for the software can have impact on the performance of the project. With some program languages it is easier to coordinate with other developers. This may influence the performance of the project (Plenert & Mason, n.d.). Eight dummy variables are included for program languages, which are: 'Java' 'C++' , 'PHP', 'C', 'Python', 'C#', 'Javascript' and 'Perl'. A score of '1' on each of these variables means that the particular program language was used in the project. Sometimes a project scored 1 on more than one dummy variable for program language, meaning the project uses more than one program language. This means there is some overlap and can also explain why the total number of languages used is larger than the total sample.

3.4.3 Type of users

The performance of the project, measured in the number of downloads, can be dependent of the type of users for which the project is supposed to be. Expected is that the user market possibly influences the performance of the project. Some user markets are much larger than others and are therefore meant for a larger audience, making it more likely that these projects will get downloaded In total there were 23 dummy variables for the type of users. To keep the analysis manageable, only the five most named type of users are included as control variables. These four concerned the following type of users: 'Developers' 'End users/desktop' 'System administrators' and 'Advanced End Users' (11,7%).

3.5 Multiple regression models

We chose to perform our statistical analysis with multiple regression models, since all of our analyses consist of several continuous independent variables and only one dependent variable. The assumptions for multiple regressions have been tested (APPENDIX A). Some regression models have a high level of multicollinearity. This can be explained by the quadratic terms which are used in these regression models, which automatically causes for a higher multicollinearity. Another problem is caused by the fact that the distribution of residuals is not normally distributed, where this is an important assumption for regression. Because several hypotheses concern interaction-relationships, increasing relationships, quadratic relationships or a combination of them, a structural equation model will not be omitted in this study. Due to the interaction and quadratic terms, a structural equation model would make the interpretation of results unnecessarily complicated. We therefore chose to limit our analysis to multiple regression models, which we believe should generate the clearest results.

4. Results

4.1 Introduction

In this section the results of our analysis are discussed. We will start by describing the variables which are used in our analyses. Then the expected main relationships are considered, after which the several moderated relationships are discussed. At the end, a summary of the results is given.

4.2 Descriptives of the variables

4.2.1 Main variables

In table 1 the descriptives of each variable used in our analysis can be found. No missing values were reported for any variable

Table 1. Descriptives of the variables

	count	mean	sd	min	max
projectsize	30031	2.67	5.00	1	250.00
logprojectsize	30031	0.56	0.76	0	5.52
degree	30031	5.51	6.97	2	228.00
logdegree	30031	1.36	0.75	0.69	5.43
brokerage	30031	685.79	13087.45	0	1348779.00
logbrokerage	30031	0.78	2.15	0	14.11
closure	30031	84.62	91.27	0	246.75
performance	30031	1980.12	182552.70	0	3.11E+07
relperformance	30031	256.27	9847.01	0	942527.30
rellogperformance	30031	1.71	2.02	0	13.76
LANG1	30031	0.32	0.47	0	1
LANG2	30031	0.22	0.41	0	1
LANG3	30031	0.12	0.33	0	1
LANG4	30031	0.21	0.41	0	1
LANG5	30031	0.07	0.25	0	1
LANG6	30031	0.06	0.24	0	1
LANG7	30031	0.05	0.22	0	1
LANG8	30031	0.06	0.24	0	1
IA1	30031	0.53	0.50	0	1
IA2	30031	0.40	0.49	0	1
IA3	30031	0.16	0.36	0	1
IA4	30031	0.12	0.32	0	1
N	30031				

The average project size, that is to say the number of developers, is 2 to 3 developers (*M* = 2.67, *SD* = 5.00). The smallest project consists of one developer while the largest project consists of 250 developers. 49% of the OSS-projects consisted of only one developer.

The average degree of a project was found to be 5, which means a project is on average connected to 5 other projects (*M* = 5.51, *SD* = 6.97). However, the degree has a large range, since the lowest degree was found to be 2, while the highest degree was found to be 228.

The level of brokerage had a mean of 685 with a minimum of zero and maximum of 1 348 779 (*M* = 685.79, *SD* = 13087.45). The average closure in the dataset was 84, with a minimum of 0 and a maximum of 246 (*M* = 84.62, *SD* = 91.27).

The average performance, which was measured by the number of downloads in a 60-day widow, was found to be 1980, with a surprising high standard deviation (*M* = 1980.12, *SD* = 182552.70). The minimum is 0 and the maximum 31 100 000. This means the software of a project was downloaded 1980 times on average in a 60 days' time widow, but is varies quite substantially between the projects. When this performance is made relative to the number of developers within the project, the mean is 256 *(M* = 256.27, *SD* = 9847.01) with a minimum of 0 and a maximum of 942 527.30

4.2.2 Control variables

Five different project-stages were distinguished, each of them representing a certain stage in which the project is situated. The following percentages were found for the several stages: 'Developing' (28,3%), 'Alpha' (17,9%), 'Beta' (25,2%), 'Stable' (25,6%) and 'Mature' (2,4%).

The most frequently used program language was 'Java' (31,9%). Also 'C++' (22%), 'C++' (22%) and 'C' were found to be frequently used program languages. Less popular languages were 'PHP' (12%), 'Python' (6,6%), 'C#' (6,2%), 'JavaScript' (5,2%) and 'Perl' (6,3%).

More than a half of the projects focused on software developers (52,7%), also a high percentage of projects makes software meant for end desktop users (40%). The other two categories for the type of users, being 'System administrators' (15,8%) and 'Advanced End Users' (11,7%), were less common. The total of percentages of the four categories for type of users is higher than 100%. This can be explained due to the fact that it is possible that software is designated for more than one category of users.

A correlation matrix has been made to investigate whether the main variables are highly correlated to each other. This could create issues with multicollinearity when performing the multiple regressions. The correlation matrix, which can be found in table 2 shows none of the main variables are extremely highly correlated with each other. Besides the correlation between the natural variables and their logarithmic transformation, the highest correlation was found between closure and (log)degree, *r* (30 029) = .701, *p* < .001. Next, degree and project size turned out to be positively correlated to each other, *r* (30029) = .528, *p* < .001. Brokerage was also found to be positively correlated with project size, *r* (30 029) = .502, *p* < .001, as well as their logarithmic transformations, *r* (30 029) = .606, *p* < .001. Lastly, table 2 shows a moderate positive correlation between (log)brokerage and (log)degree, *r* (30029), .569, *p* < .001). All other correlations between variables were found to be lower than *r* (30 029) = .4.

Table 2. Correlation matrix for the main variables.

	projectsize	logprojectsize	degree	logdegree	brokerage	logbrokerage	closure	performance	rellogperformance
projectsize	1								
logprojectsize	0.706^{***}	1							
degree	0.528^{***}	0.394^{***}	1						
logdegree	0.338^{***}	0.382^{***}	0.815^{***}	1					
brokerage	0.501^{***}	0.184^{***}	0.496^{***}	0.177^{***}	1				
logbrokerage	0.502^{***}	0.606^{***}	0.572^{***}	0.569^{***}	0.248^{***}	1			
closure	-0.0629^{***}	-0.0618^{***}	0.368^{***}	0.701^{***}	-0.0248^{***}	-0.0413^{***}	1		
performance	0.0487^{***}	0.0320^{***}	0.0253^{***}	0.0184^{**}	0.0187^{**}	0.0185^{**}	-0.00559	1	
rellogperformance	0.0900^{***}	0.0984^{***}	0.117^{***}	0.128^{***}	0.0612^{***}	0.160^{***}	0.0309^{***}	0.0555^{***}	1

* $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$

4.3 Main relationships

4.3.1 The relationship between project size and project performance

Table 3 shows predictors for project outcome performance. The variables project size and brokerage are expected to be related to the project performance. In model 1 and 2, the relationship between these variables and project performance is tested separately. In model 3, the effects of project size and brokerage on project performance are added together. At last, in model 4, the control variables are added.

With an explained variance of less than 3%, the models 1, 2 and 3 appeared to be explaining a low percentage of the variance. Model 4, including the control variables, did explain 24,2% of the variance.

Our first hypothesis, that the relationship between project size and project performance would be inversely u-shaped, was rejected (*B* = 0.00575, *t* (30 012) = 0.41, *p* = .683), neither a normal linear relationship between project size and project performance was found (*B* = -0.0248, *t* (30 012) = -0.73, *p* = .467). Although the quadratic term of project size did seem to be significant in the first model (*B* = 0.117, *t* (30 027) = 7.97, *p* < .001) and in the third model (*B* = 0,045, *t* (30 025) = 2.8, *p* < .01), this significant relationship disappeared in model 4, once the control variables were added to the model.

Table 3. Multiple regression model for predicting project performance.

	Model 1		**Model 2**		**Model 3**		**Model 4**	
	(log)Project performance		(log)Project performance		(log)Project performance		(log)Project performance	
	B	SD	B	SD	B	SD	B	SD
Constant	1.607^{***}	0.0152	1.601^{***}	0.0124	1.609^{***}	0.0151	-0.766^{***}	0.0380
Main variables								
(log)Projectsize	0.00233	0.0358			-0.0812^{*}	0.0385	-0.0248	0.0341
(log)Projectsize²	0.117^{***}	0.0147			0.0446^{**}	0.0160	0.00575	0.0141
(log)Brokerage			0.0928^{***}	0.0191	0.105^{***}	0.0209	0.0544^{**}	0.0184
(log)Brokerage²			0.00746^{**}	0.00238	0.00501	0.00256	0.00449^{*}	0.00227
Control variables								
Status of the project							0.641^{***}	0.00728
Language: Java							-0.0259	0.0271
Language: C++							0.144^{***}	0.0275
Language: PHP							-0.237^{***}	0.0352
Language: C							0.123^{***}	0.0279
Language: Python							-0.00725	0.0421
Language: C#							0.109^{*}	0.0446
Language: Javascript							-0.0369	0.0472
Language: Perl							-0.262^{***}	0.0435
Users: Developers							0.0933^{***}	0.0221
Users: End users							0.383^{***}	0.0223
Users: System administrators							0.107^{***}	0.0290
Users: Advanced end users							0.0718^{*}	0.0318
N	30031		30031		30031		30031	
R^2	0.012		0.026		0.026		0.242	

* $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$

4.3.2 The relationship between brokerage and project performance

Brokerage proved to be a significant positive predictor for project performance, confirming hypothesis 2 (*B* = 0,054, *t* (30 012) = 2.95, p < .01). Contrary to expectation, the relationship between brokerage and performance turned out to be *increasingly* positive rather than positive linear, since the quadratic term of brokerage was significant (*B* = 0.004, *t* (30 012) = 1.98, *p* < .05). This means brokerage predicts an increase in project performance and the higher the level of brokerage of a project, the stronger the increase will be. The predicting regression equation for brokerage on the project performance can be found below, together with the graph of this equation.

$$\textit{(log)Project performance} = -0{,}766 + 0{,}054*\textit{(log)brokerage} + 0{,}004 * \textit{(log)brokerage}^2$$

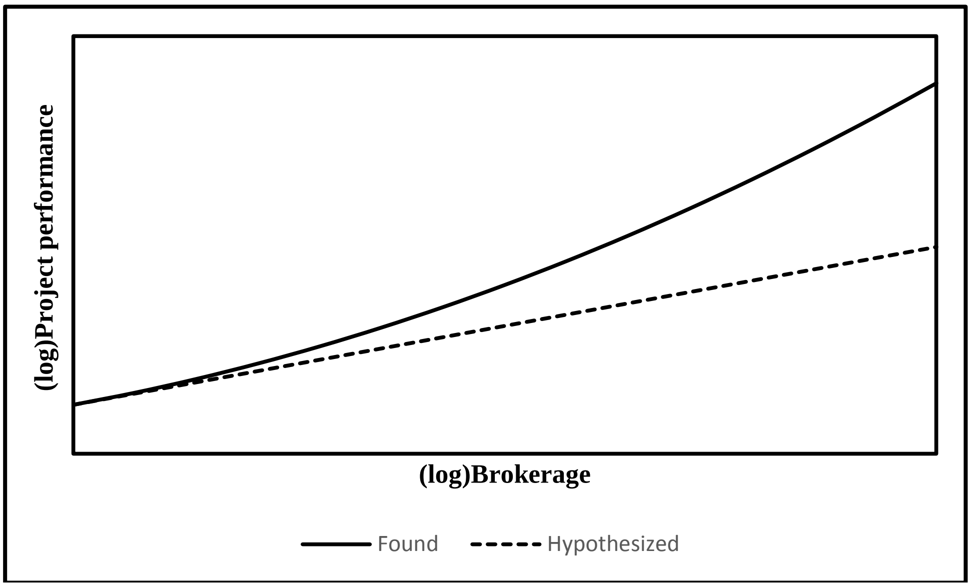

Figure 14. The increasing positive relationship between brokerage and project performance (H1 confirmed)

4.3.3 The relationship between project size and degree

Table 4. Multiple regression for predicting degree by project size

	Model 1		**Model 2**	
	Degree		Degree	
	B	SD	B	SD
Constant	3.395***	0.0421	1.899***	0.125
Main variables				
Project size	0.799***	0.00992	0.774***	0.00999
Project size^2	-0.000728***	0.0000821	-0.000620*** **(H3)**	0.0000818
Control variables				
Status of the project			0.329***	0.0242
Language: Java			0.108	0.0903
Language: C++			-0.275**	0.0915
Language: PHP			-0.457***	0.117
Language: C			0.790***	0.0929
Language: Python			0.505***	0.141
Language: C#			-0.427**	0.149
Language: Javascript			0.00650	0.158
Language: Perl			-0.0467	0.145
Users: Developers			0.680***	0.0735
Users: End users			0.117	0.0746
Users: System administrators			-0.0550	0.0970
Users: Advanced end users			-0.413***	0.106
N	30031		30031	
R^2	0.280		0.292	

* $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$

Table 4. shows the results for the expected relationship between project size on degree (H3). In this regression analysis, the 'natural' variables are used rather than their logarithmic transformations. This is done because our hypothesis about the relationship between project size and degree is based on simulations. Simulations try to simulate the real world as much as possible, meaning that it is not necessary to fulfill statistical methods to verify the outcome of the simulations. Because of this, the simulation on which we based our hypothesis, did not use the logarithmic transformation. It would be impossible to verify our hypothesis when we would work with variables which were exposed to logarithmic transformations, since then we would be comparing two very different things.

In the first model degree is predicted by project size and the quadratic term of project size. The model explains a high percentage of the explained variance ($R^2 = .28$, F (2, 30 028) = 5847.12, $p < .001$). Model 2, in which the control variables are added, appears to add little to explained variance ($R^2 = .26$, F (15, 30 015) = 824.95, $p < .001$).

In model 1 the normal term as well as the quadratic term of project size are significant ($B = 0.799$, t (30 027) = 80.60, p< .001 and B = - 0.000728, t (30 027) = -8.87 $p < .001$). The normal term for project size appears to be positive and the quadratic term negative. This implies that project size is a decreasing positive predictor for degree. The found relationship does almost not change when the control variables are added in the second model ($B = 0.774$, t (30 015) = 77.49, $p < .001$ and B= 0.00062, t (30 015) = 7.59 $p < .001$). Because both the coefficient and the quadratic term for project size turned out to be significant predictors for degree, we can say project size is a decreasing positive predictor for degree. This confirms hypothesis 3. The following regression equation can be defined for predicting degree by project size.

$$Degree = 1.899 + 0.774*Projectsize - 0.00062*Projectsize^2$$

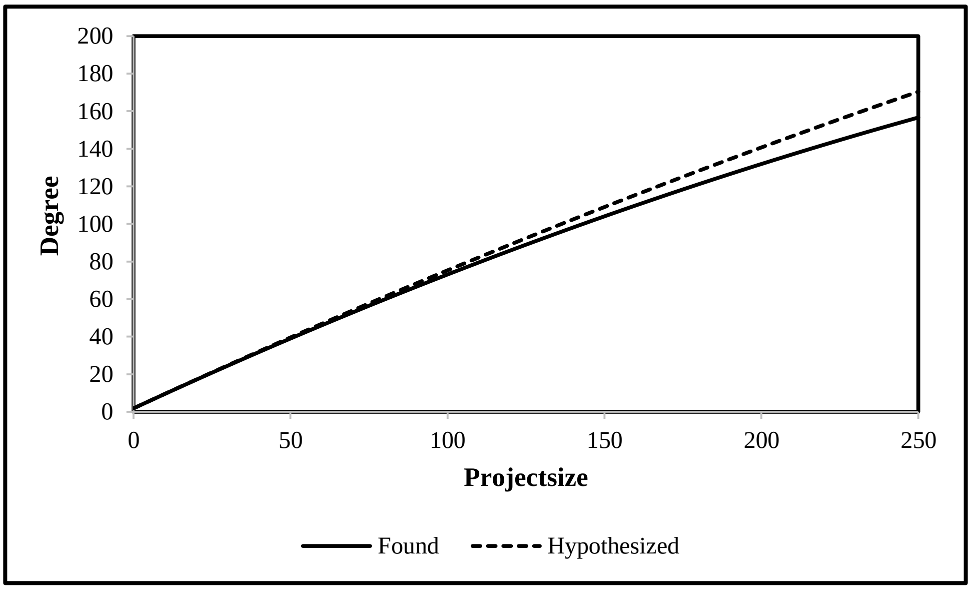

Figure 15. The decreasing positive relationship between project size and degree (H3 confirmed)

4.3.4 The relationship between degree and brokerage

Table 5. shows the relationship between degree and brokerage (H4) and between degree and closure (H5). Due the same reason as for the relationship between project size and degree, the natural variables have been used, instead of their logarithmic transformations. Model 1 and 2 test the increasing positive relationship between degree and brokerage, by using a quadratic term. Also the cubed term of degree was added to test whether there could be a cubed relationship between degree and brokerage. Model 1 tests the relationship between degree and brokerage. Model 1 turned out to explain a high percentage of the variance ($R^2 = .684$, F (3, 30027) = 21.691, $p < .001$). By adding the control variables in model 2, the explained variance barely increased by 0,02% ($R^2 = .6862$, F (16, 30014) = 4101.66, $p < .001$). We choose deliberately for not including closure as a control variable. It is not necessary to add closure as a control variable because it is not in a causal relation with brokerage. The two are dependent of each other but one does not cause the other specifically.

Table 5. Multiple regression for predicting brokerage and closure by degree

	Model 1		**Model 2**		**Model 3**		**Model 4**	
	Brokerage		Brokerage		Closure		Closure	
	B	SD	B	SD	B	SD	B	SD
Constant	1327.7***	70.64	426.1**	160.7	37.89***	0.666	44.11***	1.714
Main variables								
Degree	-426.4***	13.76	-450.0***	13.93	9.621***	0.100	9.625***	0.102
Degree^2	21.12***	0.363	21.47***	0.363	-0.0790***	0.00124	-0.0789***	0.00124
Degree^3	0.0112***	0.00171	0.00995***	0.00172				
Control variables								
Status of the project			151.7***	30.38			0.374	0.329
Language: Java			-151.4	112.4			-1.687	1.219
Language: C++			175.4	114.0			-6.543***	1.236
Language: PHP			-353.1*	146.6			-11.22***	1.589
Language: C			661.3***	116.1			-6.781***	1.259
Language: Python			1053.7***	175.5			-2.388	1.903
Language: C#			110.4	186.0			-9.616***	2.017
Language: JavaScript			-594.6**	196.7			-2.307	2.133
Language: Perl			227.1	181.1			-2.418	1.964
Users: Developers			437.6***	91.99			-0.854	0.997
Users: End users			267.7**	93.10			-3.389***	1.010
Users: System admin			40.16	121.1			-0.501	1.313
Users: Advanced end users			12.24	132.6			0.992	1.438
N	30031		30031		30031		30031	
R^2	0.684		0.686		0.238		0.241	

* $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$

Like model 1, model 2 shows degree is a significant negative predictor for brokerage (B = -450, t (30013) = -32.31, $p < .001$). However, model 2 shows as well a significant positive coefficient for the quadratic term of degree (B = 21.47, t (30013) = 59.08, $p < .001$) and a significant positive coefficient for the cubed term of degree (B = 0.00995, t (30013) = 5.80, $p < .001$), implying degree is an increasing positive predictor for brokerage expect for the lowest levels of degree. This means that degree is, only at the real first instance, a negative predictor for brokerage. However, when degree gets higher it soon becomes an increasingly positive predictor for brokerage. Therefore hypothesis 4 is confirmed, stating that degree is an increasing positive predictor for brokerage. Only for the lowest values, degree is a negative predictor for brokerage. The regression equation for predicting brokerage by degree can be found below.

$$Brokerage = 352.4 - 465*Degree + 21.87*Degree^2 + 0.0083*Degree^3$$

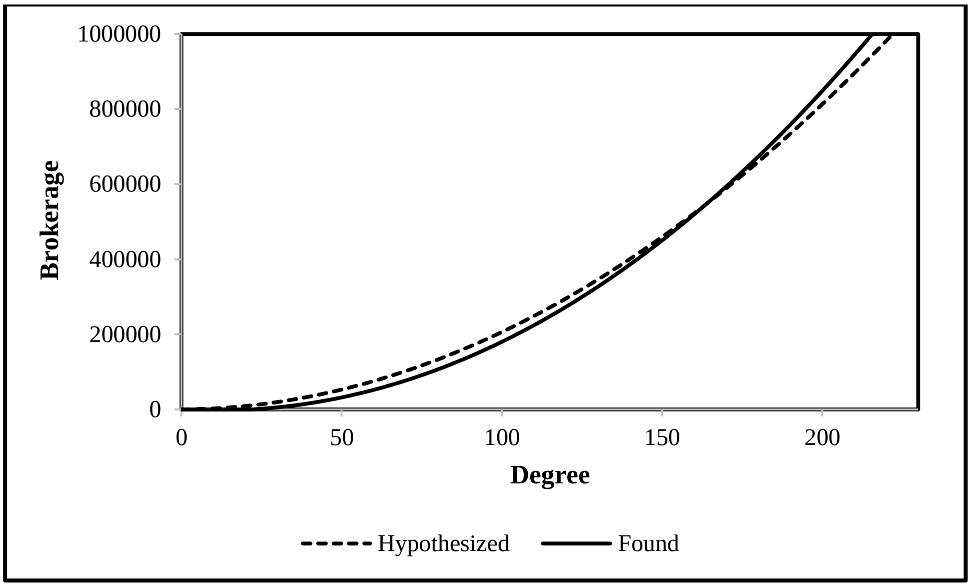

Figure 16. The increasing positive relationship between brokerage and project performance (H4 confirmed)

4.3.5 The relationship between degree and closure

In model 3 and 4 of table 5, the relationship between degree and closure is tested. Once again the natural variables are used instead of the logarithmic transformations, due to the same reasons as discussed before. In model 3, only the relationship between degree and closure was examined. In the fourth model, the control variables are added. Based on the same considerations as in former paragraph, we did not choose to add brokerage as a control variable for predicting closure by degree. A quadratic term and cubed term of degree are added to test for non-linear relationships between degree and closure. Both models explained a high percentage of the variance, which was 23.81% for model 3 (R^2 = .261, F (2, 30028) = 4692.52, p < .001) and 24.11% for the second model (R^2 = .2411, F (15, 30015) = 635.73, p < .001).

In model 4 the normal term of degree is a significant positive predictor for closure (B =9.625, t (30014) = 94.67, p < .001). However, the quadratic term for degree was found to be a significant negative predictor for closure (B = -0.0789, t (30014) = -63.47, p < .001). This implies a nonlinear relationship between degree and closure, in which degree first is a positive predictor for closure and then at a certain point will turn to an increasing negative predictor. Hypothesis 5 predicted that degree would be a decreasing negative predictor for closure. This makes hypothesis 5 not confirmed. Another finding in model 4 is the significant quadratic term of brokerage. This implies that brokerage is an increasing positive predictor for closure, which was not expected (*B* = 1.44e-09, *t* (30011) = 17.87, *p* < .001). This does not correspond with the earlier found inversed u-shaped relationship between brokerage and closure.

The formula for predicting closure by degree can be defined as follows:

$Closure = 23.62 + 16.53{*}Degree - 0.296{*}Degree^2 + 0.00111{*}Degree^3$

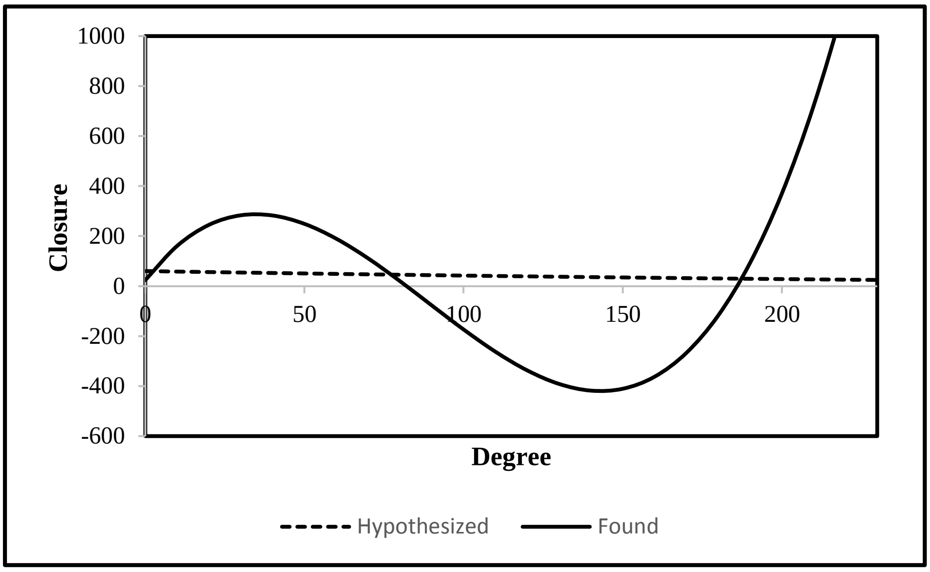

Figure 17. The quadratic relationship between brokerage and project performance (H5 rejected)

Since the regression equation for predicting closure by degree was not satisfying, we explored the cause of this. Once a plot of the relation between degree and closure was created it clearly showed a decreasing negative relation, as predicted in our hypothesis 5. However, for a curious reason there are found very low levels of closure for the first values of degree. This very low values - most of them have zero closure - a positive relationship between degree and closure is predicted by the regression. The reason for the low closure for the lowest levels of degree is uncertain. A reason might be that this is caused by small projects, existing of 1 to 5 developers, which all have one side project on which they are working alone.

Because the regression line is biased by this low values of degree create some bias in the regression line we chose to perform another regression which omitted the first six values for degree. By only taking projects with a higher degree than 6 there remained 6400 projects. The results can be found in table 6.

The model in table 6 explained a variance of 7.78% (R^2 = .078, *F* (15, 6384) = 35, *p* < .001). The model shows that the normal term of degree is a significant negative predictor for closure (*B* = -1.220, *t* (6383) =

Table 6. Multiple regression for predicting brokerage and closure by degree

	Model 1	
	Closure	
	B	SD
Constant	227.5^{***}	3.722
Main variables		
Degree	-1.220^{***}	0.149
Degree²	0.00314^{*}	0.00138
Control variables		
Status of the project	-5.882^{***}	0.653
Language: Java	-8.655^{***}	2.266
Language: C++	-14.10^{***}	2.323
Language: PHP	-9.124^{**}	3.341
Language: C	-17.34^{***}	2.261
Language: Python	-19.04^{***}	3.501
Language: C#	-8.453^{*}	4.197
Language: JavaScript	-2.414	4.155
Language: Perl	-10.53^{**}	3.653
Users: Developers	-14.77^{***}	1.952
Users: End users	-12.85^{***}	1.942
Users: System admin	-5.070^{*}	2.496
Users: Advanced end users	11.33^{***}	2.958
N	6400	
R^2	0.078	

$^{*}p < 0.05$, $^{**}p < 0.01$, $^{***}p < 0.001$

-8.2, *p* < .001). The quadratic term of degree appears to be positive (*B* = 0.00314, *t* (30031) = 2.28, *p* < .05). This implies a decreasing negative relationship between degree and closure. For that reason hypothesis 5 can ultimately be confirmed, although the decreasing negative relationship only applies for projects with a higher degree than six. The following regression formula can be given for predicting closure by degree for projects with a higher degree than 6:

$Closure = 227.5 – 1.22*Degree + 0.00314*Degree^2$

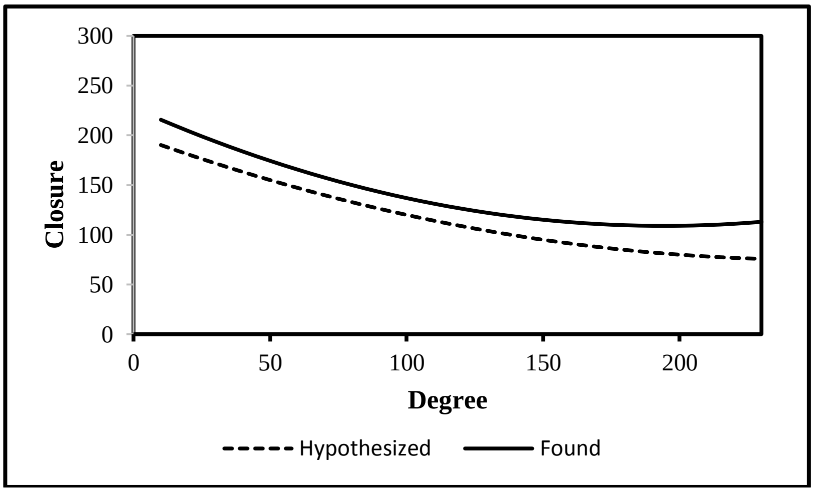

Figure 18. The quadratic relationship between brokerage and project performance (H5 confirmed)

4.4 Moderated relationships

4.4.1 Closure moderating the positive relationship between brokerage and project performance

Table 7 is used for testing hypothesis 6 about the possible interaction between brokerage and closure for the relationship between brokerage and project performance. Model 1 tests the hypothesis without the control variables. The model appears to explain a poor percentage of the variance, namely 3% (R^2 = .03, F (8, 30022) = 114.27). This implies that brokerage and closure are not the main predictors for project performance.

Table 7. Multiple regression for predicting project performance by brokerage and closure

	Model 1		**Model 2**		**Model 3**	
	(log)Project performance		(log)Project performance		(log)Project performance	
	B	SD	B	SD	B	SD
Constant	1.517***	0.0189	-0.787***	0.0388	-0.628***	0.0456
Main variables						
(log)Projectsize	-0.0566	0.0388	-0.00438	0.0344		
(log)Projectsize^2	0.0256	0.0165	-0.0128	0.0146		
(log)Brokerage	0.170***	0.0248	0.096 ***	.022		
(log)Brokerage^2	0.00237	0.00267	0.00333	0.00236		
Closure	0.00190**	0.000735	0.00225***	0.000649		
Closure^2	-0.00000433	0.00000363	-0.0000102**	0.00000321		
(log)Brokerage * Closure	-0.000172	0.000339	-0.000192	0.000300		
(log)Brokerage * Closure^2	-0.00000376*	0.00000182	-0.00000295	0.00000161 **(H6)**		
(log)Projectsize (cent)					-0.0186	0.0219
(log)Projectsize^2 (cent)					-0.0128	0.0146
(log)Brokerage (cent)					0.064***	0.0178
(log)Brokerage^2 (cent)					0.00333	0.00236
Closure (cent)					-0.0000188	0.000150
Closure^2 (cent)					-0.0000125***	0.00000303
(log)Brokerage * Closure (cent)					-0.000691***	0.0000998
(log)Brokerage * Closure^2 (cent)					-0.00000295	0.00000161
Control variables						
Status of the project			0.640***	0.00730	0.640***	0.00730
Language: Java			-0.0253	0.0271	-0.0253	0.0271
Language: C++			0.145***	0.0274	0.145***	0.0274
Language: PHP			-0.236***	0.0352	-0.236***	0.0352
Language: C			0.123***	0.0279	0.123***	0.0279
Language: Python			-0.0116	0.0421	-0.0116	0.0421
Language: C#			0.111*	0.0446	0.111*	0.0446
Language: Javascript			-0.0357	0.0472	-0.0357	0.0472
Language: Perl			-0.265***	0.0434	-0.265***	0.0434
Users: Developers			0.0923***	0.0221	0.0923***	0.0221
Users: End users			0.379***	0.0223	0.379***	0.0223
Users: System administrators			0.106***	0.0290	0.106***	0.0290
Users: Advanced end users			0.0720*	0.0318	0.0720*	0.0318
N	30031		30031		30031	
R^2	0.030		0.243		0.243	

* $p < 0.05$, ** $p < 0.01$, *** $p < 0.00$

Model 2, in which the control variables are included, explains a more reasonable percentage of the variance, which is 24.3% ($R^2 = .243$, F (21, 30009) = 459.034, $p < .001$). At first, model 2 shows again brokerage to be a positive predictor for project performance ($B = .096$, t (30008) = 4.4, $p < .001$). However, in contrast to our earlier findings in table 3, there is no evidence that brokerage is an *increasing* positive predictor for project performance. Closure appears to be increasingly negative related to project performance ($B = -0.0000102$, t (30008) = 3.18, p < .05).

The linear interaction term between brokerage and closure is found to be insignificant while the quadratic interaction term between brokerage and closure is found to be slightly significant (B = -2,952, t (30008) = -1.83, $p = .067$). The negative coefficient of the quadratic interaction term between brokerage and closure implies the found positive relationship between brokerage and project performance is negatively moderated by closure. Although the p-value for this coefficient did just not reach the significance level ($p = 0.067$) it can still be seen as evidence that the positive relation between brokerage and project performance mitigates as closure gets higher. This finding rejects hypothesis 6, which stated that the moderation of closure on the positive relationship between brokerage and project performance would be inversely u-shaped. Hypothesis 6 assumed a certain optimum of closure, for which the relationship between brokerage and project performance would be the most positive. To find hypothesis 6 confirmed the linear interaction term would have to be positive while the quadratic interaction term would be negative. Looking at the results we can conclude this is not the case. There is no certain level of closure that is the optimum, meaning that the optimum of closure is zero. This means for projects with low closure, the relationship between brokerage and project performance is the most positive and for projects with high closure the relationship between brokerage and project performance is the least positive. The regression equation for predicting project performance by brokerage and closure can be found below.

(log)Project performance = 0.787 + 0.096**(log)Brokerage* + 0.00225**Closure* - 0.0000102**Closure²* - 0.000002952**(log)Brokerage*Closure²*

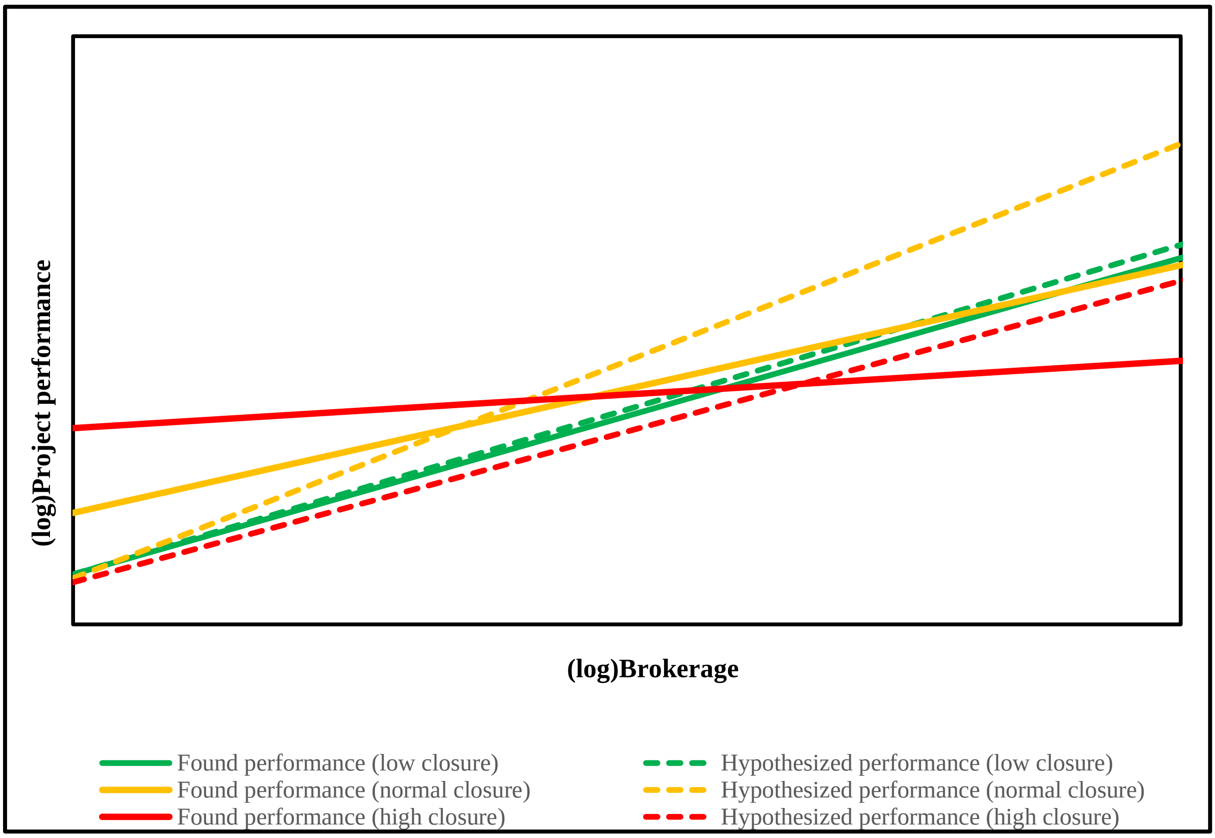

Figure 19. The negative moderation of closure on the positive relationship between brokerage and project performance (H6 not confirmed)

Because the quadratic interaction term between closure and brokerage was found to be almost significant, we also performed a regression analysis with the centered variables which are used in the quadratic terms. It could be possible the quadratic interaction term is insignificant because the high collinearity which naturally arises between the linear terms and their quadratic terms. An analysis of the VIF-values for model 2 also confirms that the linear terms are highly correlated to the quadratic terms (mean VIF = 31.84). The collinearity can be reduced when the variables which are used in the quadratic terms are transformed to centered variables, which is done by subtracting the mean of the variable from the variable.

Model 3 shows the regression analysis with the centered variables. The explained variance corresponds with the explained variance in model 2 (R^2 = .243, F (21, 30009) = 459.034, p < .001). An analysis of the VIF-values confirms that centering the variables reduces the collinearity between the linear and quadratic terms (mean VIF = 4.98). However, as can be seen in model 3, the quadratic interaction term between brokerage and closure remains insignificant ((B = -2,952, t (30008) = -1.83, p < 0.1). Yet, the normal interaction term between brokerage and closure turns out to be significant (B= -0.00069, t (30008) = -6.93, p < .001). Since brokerage has only a range from 0 until 15 this coefficient is almost of worthless importance. We have to conclude model 3 does not add any value.

4.4.2 Influence of project size on the moderation of closure with the positive relation between brokerage and project performance

Lastly, hypotheses 7 and 8 are tested in table 8. Hypothesis 7 predicted that the moderation of closure - which was found in hypothesis 6 - would be the highest for small projects and the smallest for large projects. Hypothesis 8 stated that project size moderates the positive relationship between brokerage and project performance negatively. This would mean that for large projects the positive relationship between brokerage and project performance would be the least.

Table 8. Multiple regression for predicting project performance by moderated relationships between brokerage, closure and project size

	Model 1		**Model 2**		**Model 3**	
	(log)Project performance		(log)Project performance		(log)Project performance	
	B	SD	B	SD	B	SD
Constant	1.499***	0.0202	-0.792***	0.0393	-0.605***	0.0463
Main variables						
Project size	0.0726	0.0502	0.0781	0.0444		
Project size^2 (H1)	-0.0367	0.0243	-0.0617**	0.0215		
Brokerage	0.115***	0.0334	0.0489	0.0296		
Brokerage^2	-0.00297	0.00341	-0.00173	0.00302		
Closure	0.00178*	0.000888	0.00220**	0.000785		
Closure^2	-0.00000290	0.00000438	-0.00000975*	0.00000387		
Brokerage*Closure	-8.70e-08	0.000735	0.000367	0.000649		
Brokerage*Closure^2	-0.000000391	0.00000407	-0.00000285	0.00000360		
Closure*Projectsize	-0.0000530	0.00116	-0.000271	0.00102		
Closure^2*projectsize	-0.00000232	0.00000577	0.000000609	0.00000510		
Brokerage*Projectsize	0.0422*	0.0167	0.0415**	0.0147 **(H8)**		
Closure*Brokerage*Projectsize	0.000160	0.000387	-0.0000932	0.000342		
Closure^2*Brokerage*Projectsize	-0.00000303	0.00000238	-0.00000112	0.00000211 **(H7)**		
Project size (cent)					0.0108	0.0484
Project size^2 (cent)					-0.0617**	0.0215
Brokerage (cent)					0.0710***	0.0212
Brokerage^2 (cent)					-0.00173	0.00302
Closure (cent)					0.000243	0.000184
Closure^2 (cent)					-0.0000121***	0.00000340
Brokerage*Closure (cent)					-0.000274	0.000178
Brokerage*Closure^2 (cent)					-0.00000348	0.00000269
Closure*Projectsize (cent)					-0.000390	0.000243
Closure^2*projectsize (cent)					-0.000000271	0.00000492
Brokerage*Projectsize (cent)					0.0255*	0.0126
Closure*Brokerage*Projectsize (cent)					-0.000283*	0.000134
Closure^2*Brokerage*Projectsize (cent)					-0.00000112 **(H7)**	0.00000211
Control variables						
STATUSCODE			0.640***	0.00730	0.640***	0.00730
LANG1			-0.0245	0.0271	-0.0245	0.0271

LANG2		0.145^{***}	0.0274	0.145^{***}	0.0274
LANG3		-0.237^{***}	0.0352	-0.237^{***}	0.0352
LANG4		0.123^{***}	0.0279	0.123^{***}	0.0279
LANG5		-0.0116	0.0421	-0.0116	0.0421
LANG6		0.110^{*}	0.0446	0.110^{*}	0.0446
LANG7		-0.0337	0.0472	-0.0337	0.0472
LANG8		-0.266^{***}	0.0434	-0.266^{***}	0.0434
IA1		0.0911^{***}	0.0221	0.0911^{***}	0.0221
IA2		0.378^{***}	0.0223	0.378^{***}	0.0223
IA3		0.105^{***}	0.0290	0.105^{***}	0.0290
IA4		0.0724^{*}	0.0318	0.0724^{*}	0.0318
N	30031	30031		30031	
R^2	0.030	0.243		0.243	

* $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$

To test hypothesis 7, a comprehensive interaction term between *(log)Brokerage*Closure²* and *(log)Project size* has been constructed. This results in the interaction term *(log)Brokerage*Closure²*(log)Project size*. Due to the comprehensiveness of this interaction term, it was inevitable that all lower terms had to be introduced in the regression analysis, which could cause some problems with the clarity of the regression table. To verify hypothesis 8, the interaction term *(log)Brokerage*Project size* is used.

In model 1 the interaction terms for hypothesis 7 and 8 are introduced without the control variables. With only 3% of explained variance, the model is not very powerful (R^2 = .03, *F* (13, 30017) = 72, *p* < .001). Model 2, in which the control variables were added, explained 24.3% of the variance (R^2 = .243, *F* (26, 30004) = 371.34, *p* < .001). Model 2 shows that project size does not moderate the moderation of closure on the relation between brokerage and project performance, since the term for this is far from significant (*B* = -1.12e^{-06}, *t* (30003) = 0.53, *p* = .593). This means that, independent of the project size, closure is negatively moderating the positive relationship between brokerage and project performance. Furthermore, model 2 only shows a significant negative coefficient for the quadratic term of project size (*B* = 0.0617, *t* (30003) = -2.87, *p* < .01) and the quadratic term of closure (*B* = 0.00000975, *t* (30003) = -2.52, *p* < .01), implying project size and closure to be increasing negative predictors for project performance.

As discussed in the former paragraph, the insignificance of the interaction term for hypothesis 7 could be caused by the fact that it contains a quadratic term. Therefore there has been performed a regression analysis with centered variables, which is shown in model 3. An analysis of the VIF-values shows that model 3 reduces the VIF-values to a large extent, which means the multicollinearity has decreased. Model 3 does show more significant coefficients. Nevertheless, the quadratic interaction term for hypothesis 7 still turns out to be insignificant. However, the linear interaction term does: *(log)Brokerage*Closure*Project size* turns out to be significant in model 3 (*B* = 0.000283, *t* (30003) = -

2.12, *p* < .05). This means project size does affect the negative moderation of closure on the positive relationship between brokerage and project performance. Since the coefficient of the term is negative, it means that the negative moderating effect of closure on the positive relationship between brokerage and project performance is less for small projects. In simpler words: closure is negatively moderating the positive relationship between brokerage and project size and this negative moderation is the strongest for large projects. Hypothesis 7 has therefore to be rejected since it predicted that the moderation of closure would be the least for larger projects. It appears to be the opposite. The regression equation is given below. Since hypotheses 7 draws forth on hypothesis 6, which was not confirmed, the lines for hypothesis 7 are not displayed in the graph.

$$\textit{(log)Project performance} = -0.792 - 0.0617{*}\textit{(log)Projectsize}^2 + 0.0710{*}\textit{(log)Brokerage} - 0.0000121{*}\textit{Closure}^2 - 0.000283{*}\textit{(log)Brokerage*Closure*Project size}$$

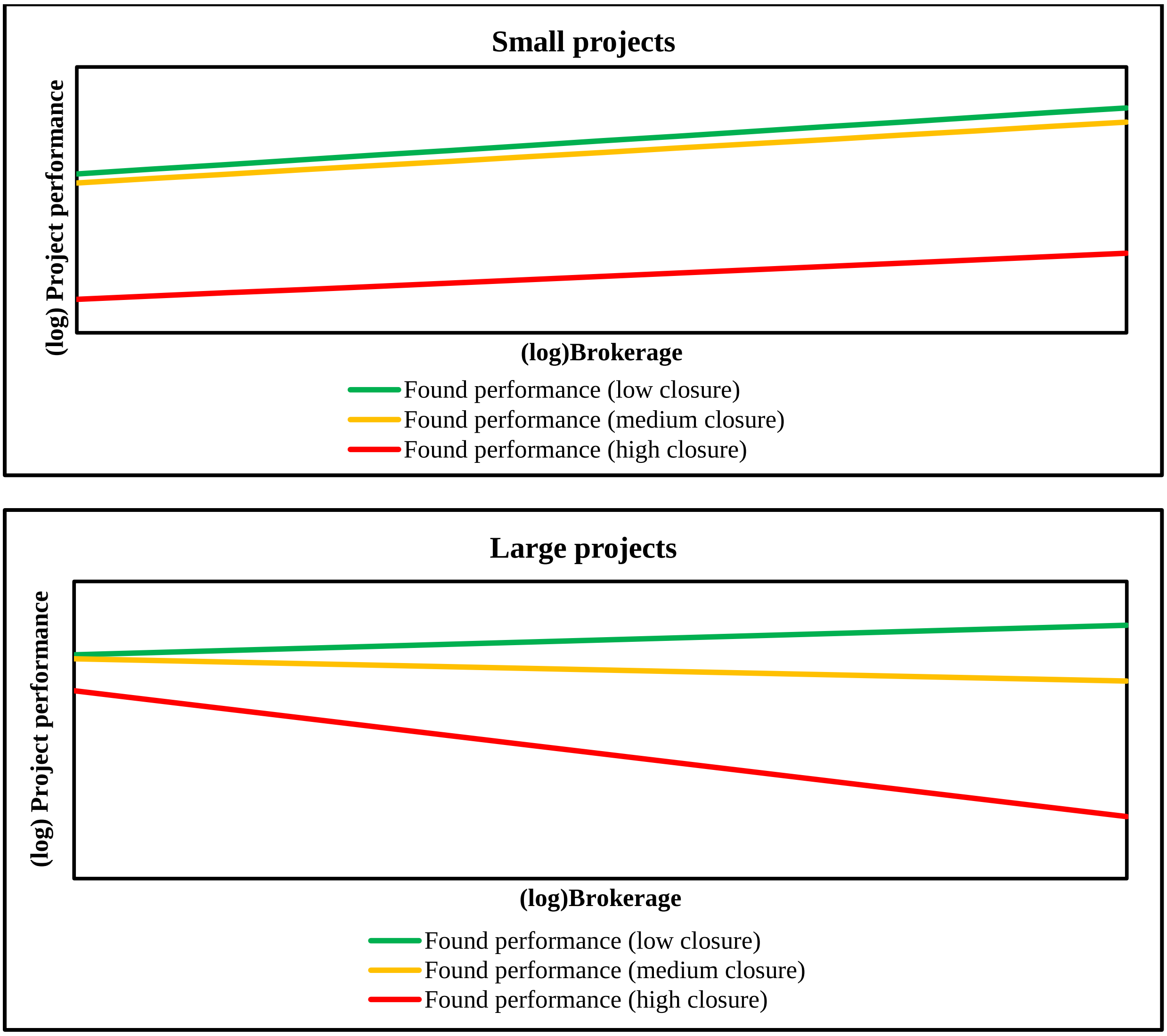

Figure 20. Project size moderating the moderation of closure on the relationship between brokerage and project performance, divided by project size.

4.4.3 Project size moderating the positive relationship between brokerage and project performance

In model 2 of table 8, hypothesis 8 is tested. Hypothesis 8 stated the positive relationship between brokerage and project performance would be the most positive for small projects and the least for large projects.

Model 2 shows a significant interaction term between brokerage and project size, which turns out to have a negative coefficient ($B = 0.0415$, $t\ (30003) = 2.82$, $p < .01$). This means project size moderates the positive relationship between brokerage and project performance positively. For larger projects the positive relationship between brokerage and project performance is more positive. This rejects hypothesis 8, since it predicted that for larger projects brokerage, would be a *smaller* positive predictor for project performance.

The quadratic term between project size and brokerage appeared to be insignificant. Therefore the moderation of project size on the relationship between brokerage and project performance is linear. The regression equation can be found below.

$$\textit{(log)Project performance} = -0.792 - 0.0617{*}\textit{(log)Projectsize}^{2} + 0.0415{*}\textit{(log)Brokerage}{*}\textit{(log)Projectsize}$$

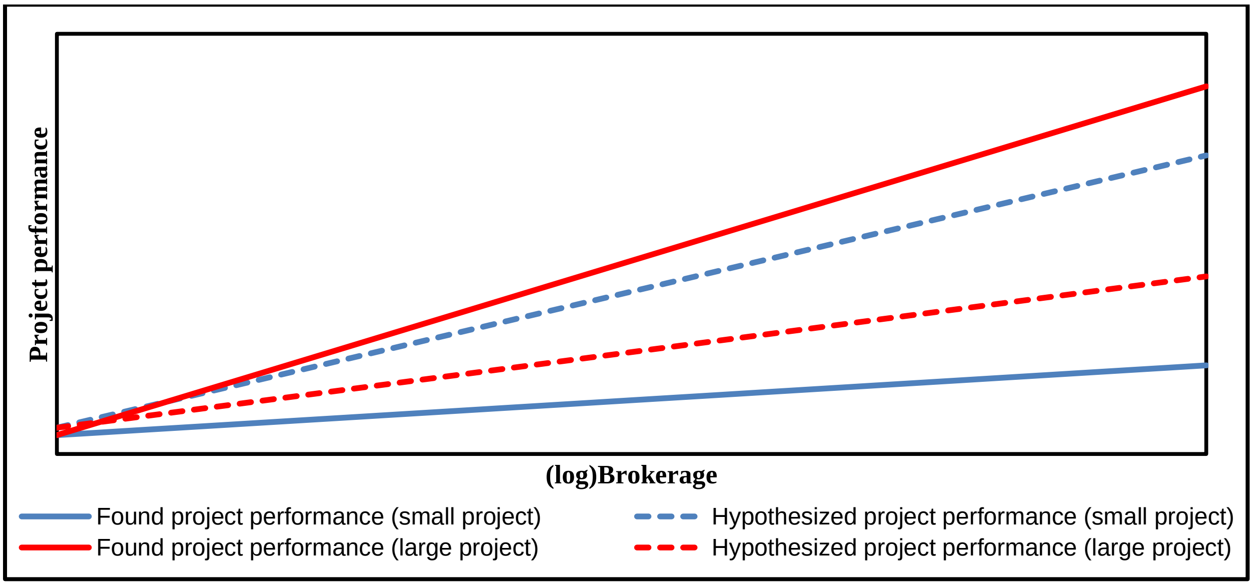

Figure 19. The positive moderation of project size on the moderation of closure on the positive relationship between brokerage and project performance (H8 rejected)

5. Conclusion

5.1 Aim of the study

This study investigated the project performance in the OSS-community, by looking to project size and social network characteristics of the project. Yet little is known of the certain ways in how project size could influence the performance of the project. Nevertheless, knowledge about the impact of project size on project performance is very useful for planning in all levels of management, such as strategical management as well as management on the floor. Knowledge about how project size could influence project performance via social networks perfectly assembles two major research fields, which are research about the influence of group size on the group performance and the research field of social networks. By focusing on the interaction between project size and its social network characteristics we hoped to give more insight in the ways project size could influence project performance. First was investigated how project size directly influences the project performance. Then the influence of project size on social networks was examined. At last, the moderating effect of project size on the relationship between social network characteristics and project performance was explored.

5.2 Findings

5.2.1 Predicting project performance by project size and brokerage

For predicting performance project size did not appear to be a significant predictor. Although in some of our models effects of project size on project performance were found, these effects did not persist and were contradicting to each other. Therefore, hypothesis 1 - which stated that the relationship between project size and project performance would be inversed u-shaped - is rejected. In contrast to project size, brokerage appeared to be an increasing positive predictor for project performance. As the project size increases the project performance becomes better and this improvement is stronger as the project size increases more. This confirmed our hypothesis 2 which was grounded on Burt’s theory about structural holes. At last, although no hypothesis was formulated on this, closure appeared to be mostly negatively related to project performance.

5.2.2 Predicting social network characteristics by project size

The relation between project size and social network characteristics, being brokerage and closure, was expected to go via degree, which means the number of ties. Confirming hypothesis 3 it was found that project size is decreasing positively related to degree. The more developers working on a project, the more ties arise to other projects, although the increase of ties decreases a little bit when the project reaches a certain number of developers. This can be explained by the fact that it becomes harder for a project with many ties to acquire *new* ties to *new* project since it is already connected with so many projects.

On its turn, degree affects the brokerage of the project, confirming our hypothesis 4. Degree appeared to be an increasing positive predictor for brokerage, which can be explained by our theory based on logical assumptions.

Thus, larger projects have a higher degree (H3) and are therefore able to gain more brokerage (H5). The reason for this is -according to our theory - that larger projects have a higher degree. Because the higher degree they are better able to fulfill a broker position between more other projects than small projects. The possibilities of brokering increase fast since one extra tie could lead to several new broker possibilities.

Degree also appeared to be a significant negative predictor for closure, although this only applied for projects with a higher degree than 6. This confirms our hypothesis 5, which was based on a mathematical formula. No explanation could be found why projects with a degree of less than six have in general so little closure. Thus, larger projects in general do have a higher degree (H3) and therefore have lower closure (H5).

5.2.3 Moderations between brokerage, closure and project size

Several possible moderations between brokerage, closure and project size were tested. First was investigated how project size could affect the positive relationship between brokerage and project performance. In contrast to our expectations we found that closure is increasing negatively moderating the positive relationship between brokerage and project performance. This means that the relation between brokerage and project performance is the most positive for projects with low closure and the least positive for projects with high closure. In other words: to profit the most from the positive impact of brokerage a project needs low closure. This contradicts our expectations since we expected that too little as well as too much closure would be disruptive for the positive relationship between brokerage and project performance. According to our theory the relationship between brokerage and project performance would be the most positive for a medium level of closure. Apparently this is not the case.

Furthermore it was found that project size influences this negative moderation of closure mentioned above. For larger projects the negative moderation of closure on the relationship between brokerage and project performance is more negative. This implies two things. First of all, larger projects will be able to profit more from their brokerage, but this will be less when the level of closure is high. Second, smaller projects will be less able to profit from a broker position but the positive relation between brokerage and project performance will persist when closure increases. This rejects our theory that smaller projects are more sensitive for the social network structure in which they are situated. A possible explanation for this is hard to give. One could think of larger projects are better able to channel the overflow of new information.

At last it was investigated how project size could influence the positive relationship between brokerage and project performance. It was found that projects size positively moderates the positive relationship between brokerage and project performance. For larger projects the relationship appears to be more positive than for smaller projects. We could say larger projects profit more from their brokerage than small projects do. This is in contrast to what we expected since we expected especially smaller project would take advantage from brokerage since they have a lack of information inside the project.

To summarize the findings we could say at first that project size does not directly affects project performance (H1) and brokerage does affect project performance increasingly positive (H2).

However, project size appears to be a valid predictor for the social network characteristics brokerage and closure. Project size is positively related to degree (H3) and degree is increasingly positive related to brokerage (H4). Since brokerage is positively related to project performance, one could state that project size is, through brokerage, a positive predictor for project performance.
On the other hand degree - which increases as project size increases - appeared to be a negative predictor for closure (H5). Since closure turned out to be a negative predictor for project performance, it could be stated that project size is, through closure, a positive predictor for project performance. At last, project size turned out to affect the relationship between brokerage, closure and project performance. Closure negatively moderates the positive relation between brokerage and project performance (H6). Smaller projects are more resistant for the negative effect of high closure (H7). That means small projects still profit from brokerage while closure is high while large projects do less. At last, there was found that the relationship between brokerage and project performance is the strongest for large projects (H8).
Thus, project size affects project performance, through social network characteristics as well as moderates the influence of social network characteristics on project performance.

5.3 Discussion and further research

Although as much as possible was done to perform a outstanding study, a few comments have to be made about the weaknesses of this study.
First of all, the logical model behind the gathered data could be problematic. The units of analysis concerned projects instead of human people. One could wonder if the social network theories - which mostly are concentrated on human behavior – also apply on such a communities as projects.

Another problem with this study is the external validity. Although the found results are possibly highly applicable on the OSS-community it is likely that the found results are hardly of any importance outside the OSS-community. Because the theory and the underlying assumptions were fully focused on the OSS-community it is hard to generalize this to other fields of conduct. However, because this study fully applies on the OSS-community this could also be seen as an advantage. Our study contributes to solve problems which adhere specifically to the OSS-community.

Another shortcoming is the fact that there has been no distinction made between process performance and outcome performance. The mechanism could work differently for these two kind of performance measures.

Although there have been found several interesting results, further research is recommended. Further research is needed on the relationship between degree and closure. It is yet unknown why the relation is polynomial, that is to say it first increases and then decreases. Also the role of project size on the relationship between brokerage and project performance could be further investigated. Since we could not explain why larger projects are better able to convert their brokerage to outcome performance.

Literature

Allcott, B. H., Karlan, D., Möbius, M. M., Rosenblat, T. S., & Szeidl, A. (2007). Community size and network closure. *The American Economic Review*, *97*(2), 80-85.

Archetti, M. (2009). The volunteer's dilemma and the optimal size of a social group. *Journal of Theoretical Biology*, *261*(3), 475-480.

Burt, R. S. (1997). The contingent value of social capital. *Administrative science quarterly*, 339-365.

Burt, R. S. (2000). The network structure of social capital. *Research in organizational behavior*, *22*, 345-423.

Burt, R. S. (2005). *Brokerage and closure: An introduction to social capital*. OUP Oxford.

Coleman, J. S. (1988). Social capital in the creation of human capital.*American journal of sociology*, S95-S120.

Crowston, K., Annabi, H., & Howison, J. (2003). Defining open source software project success. *ICIS 2003 Proceedings*, 28.

Daley, R. C. (1978). The role of team and task characteristics in R&D team collaborative problem solving and productivity. *Management Science*, *24*(15), 1579-1588.

Daniel, S. L., & Diamant, E. I. (2008). Network Effects in OSS Development: The Impact of Users and Developers on Project Performance. *ICIS 2008 Proceedings*, 122.

El Emam, K., & Koru, A. G. (2008). A replicated survey of IT software project failures. *Software, IEEE*, *25*(5), 84-90.

Everett, M., & Borgatti, S. P. (2005). Ego network betweenness. *Social networks*, *27*(1), 31-38.

Evers, S. (2000). An introduction to Open Source software development. *Technische Universität Berlin, Fachbereich Informatik, Fachgebiet Formale Modelle, Logik und Programmierung (FLP)*.

Howison, J., Conklin, M., & Crowston, K. (2006). FLOSSmole: A collaborative repository for FLOSS research data and analyses. *International Journal of Information Technology and Web Engineering, 1(3), 17–26.*

Fowler, J. H., Dawes, C. T., & Christakis, N. A. (2009). Model of genetic variation in human social networks. *Proceedings of the National Academy of Sciences*, *106*(6), 1720-1724.

Freeman, L. C. (1978). Centrality in social networks conceptual clarification.*Social networks*, *1*(3), 215-239.

Ganley, D., & Lampe, C. (2009). The ties that bind: Social network principles in online communities. *Decision Support Systems*, *47*(3), 266-274.

Guimera, R., & Amaral, L. A. N. (2004). Modeling the world-wide airport network. *The European Physical Journal B-Condensed Matter and Complex Systems*, *38*(2), 381-385.

Granovetter, M. S. (1973). The strength of weak ties. *American journal of sociology*, 1360-1380.

Hansen, M. T., Podolny, J. M., & Pfeffer, J. (2001). So many ties, so little time: A task contingency perspective on corporate social capital in organizations. *Research in the Sociology of Organizations*, *18*(18), 21-57.

Homscheid, D., & Schaarschmidt, M. (2016, May). Between organization and community: investigating turnover intention factors of firm-sponsored open source software developers. In *Proceedings of the 8th ACM Conference on Web Science* (pp. 336-337). ACM.

Ingham, A. G., Levinger, G., Graves, J., & Peckham, V. (1974). The Ringelmann effect: Studies of group size and group performance. *Journal of Experimental Social Psychology*, *10*(4), 371-384.

Igarashi, T., Kashima, Y., Kashima, E. S., Farsides, T., Kim, U., Strack, F., ... & Yuki, M. (2008). Culture, trust, and social networks. *Asian Journal of Social Psychology*, *11*(1), 88-101.

Isaac, R. M., Walker, J. M., & Williams, A. W. (1994). Group size and the voluntary provision of public goods: experimental evidence utilizing large groups. *Journal of public Economics*, *54*(1), 1-36.

Krackhardt, D., & Stern, R. N. (1988). Informal networks and organizational crises: An experimental simulation. *Social psychology quarterly*, 123-140.

Lind, M. R., & Culler, E. (2013). Information technology project performance: The impact of critical success factors. *Perspectives and Techniques for Improving Information Technology Project Management*, 39.

Liu, J. Y. C., Chen, H. G., Chen, C. C., & Sheu, T. S. (2011). Relationships among interpersonal conflict, requirements uncertainty, and software project performance. *International Journal of Project Management*, *29*(5), 547-556.

Madey, G., Freeh, V., & Tynan, R. (2002). The open source software development phenomenon: An analysis based on social network theory.*AMCIS 2002 Proceedings*, 247.

Martin, N. L., Pearson, J. M., & Furumo, K. (2007). IS project management: Size, practices and the project management office. *Journal of Computer Information Systems*, *47*(4), 52-60.

Marsden, P. V. (2002). Egocentric and sociocentric measures of network centrality. *Social networks*, *24*(4), 407-422.

Martin, N. L., Pearson, J. M., & Furumo, K. (2007). IS project management: Size, practices and the project management office. *Journal of Computer Information Systems*, *47*(4), 52-60.

Park, S. H., & Luo, Y. (2001). Guanxi and organizational dynamics: Organizational networking in Chinese firms. *Strategic management journal*,*22*(5), 455-477.

Plenert, G., & Mason, W. H. (n.d.). OBJECT-ORIENTED PROGRAMMING. Retrieved June 04, 2016, from http://www.referenceforbusiness.com/management/Ob-Or/Object-Oriented-Programming.html

Rothenberg, R. B., Potterat, J. J., Woodhouse, D. E., Darrow, W. W., Muth, S. Q., & Klovdahl, A. S. (1995). Choosing a centrality measure: epidemiologic correlates in the Colorado Springs study of social networks.*Social Networks*, *17*(3), 273-297.

Sauer, C., Gemino, A., & Reich, B. H. (2007). The impact of size and volatility on IT project performance. *Communications of the ACM*, *50*(11), 79-84.

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), 6.

Singh, P. V., Tan, Y., & Mookerjee, V. (2008). Network effects: The influence of structural social capital on open source project success. *Management Information Systems Quarterly, Forthcoming*.

Sourgeforge (2016). Retrieved june 04, 2016, from https://sourceforge.net/.

Steinmacher, I., Conte, T. U., Treude, C., & Gerosa, M. A. (2016, May). Overcoming open source project entry barriers with a portal for newcomers. In *Proceedings of the 38th International Conference on Software Engineering*(pp. 273-284). ACM.

Stuart, T. E., & Podolny, J. M. (1999). Positional consequences of strategic alliances in the semiconductor industry. *Research in the Sociology of Organizations*, *16*(1), 161-182.

Valente, T. W., Coronges, K., Lakon, C., & Costenbader, E. (2008). How correlated are network centrality measures?. *Connections (Toronto, Ont.)*,*28*(1), 16.

Wallmark, J. T., Holmqvist, H. E. S., Eckerstein, S., & Langered, B. (1973). The increase in efficiency with size of research teams. *Engineering Management, IEEE Transactions on*, (3), 80-86.

Wallmark, J. T., & Sellerberg, B. (1966). Efficiency vs. size of research teams.*Engineering Management, IEEE Transactions on*, (3), 137-142.

Westra, I.L., Broek, E.P.H. van den (2015). The influence of institutions on organizational performance via network structure. Utrecht University, Faculty of Social and Behavioural Sciences Theses.

APPENDIX A

```
* 1. PREPARING THE DATA

* Loading data

** use "C:\Users\HansB\Dropbox\UU\Jaar 4\Blok 4\Bachelor Thesis Sociology/dataset.dta"

** use "\\soliscom.uu.nl\uu\Users\4009371\Bachelor Thesis Sociology\dataset.dta"

* Create time variable for Durbin-Watson test:

gen time=_n

tsset time

* Install esttab

ssc install estout

* for every regression the assumptions are tested:

* TEST FOR INDEPENDENCE OF RESIDUALS:
* dwstat (Durbin–Watson statistic)

* TEST FOR HETEROSKEDASTICITY:
* rvfplot
* estat hettest (Breusch–Pagan test)

* TEST FOR MULTICOLLINEARITY:
* vif

* TEST FOR OUTLIERS
* scatterplots are used
* Rename all our variables

rename NDEVELOPERS projectsize

rename BETWEENNESS5 brokerage

rename TRANSITIVITY closure

rename DEGREE degree

rename T60Downloads performance


* PREPARING PROJECTSIZE

su projectsize

tab projectsize

* No missing values

histogram projectsize

* According to the histogram the variable projectsize is very highly right-skewed.
* Therefore a log-variable has to be created.

gen logprojectsize = log(projectsize)

histogram logprojectsize

* The variable logbrokerage is less rightly skewed than the variable brokerage



* PREPARING BROKERAGE
```

```
su brokerage

tab brokerage

* No missing values

histogram brokerage

* According to the histogram the variable brokerage is very highly right-skewed.
* Therefore a log-variable has to be created. As we cannot derive a logarithm from 0 we need to add 1 to brokerage

gen logbrokerage = log(brokerage + 1)

histogram logbrokerage

* The variable logbrokerage is less rightly skewed than the variable brokerage



* PREPARING CLOSURE

su closure

tab closure

* No missing values

* Closure has a very small range. To solve this problem we multiply closure by a thousand:

gen closure_1000 = closure*1000

* PREPARING DEGREE

su degree
```

```
tab degree

* No missing values

histogram degree

* According to the histogram the variable degree is very highly right-skewed.
* Therefore a log-variable has to be created.

gen logdegree = log(degree)

histogram logdegree

* The variable logbrokerage is less rightly skewed than the variable brokerage



*PREPARING PERFORMANCE

su performance

tab performance

* Making performance relative to the projectsize

gen relperformance = performance/projectsize

histogram relperformance

su relperformance

* According to the histogram the variable relperformance is very highly right-skewed.
* Therefore a log-variable has to be created. As we cannot derive a logarithm from 0 we need to add 1 to relperformance.

gen rellogperformance = log(relperformance + 1)
```

```
histogram rellogperformance

* The variable rellogperformance is less rightly skewed than relperformance

* Checking whether its a problem we also include the projectsize in the logarithm.
* Now we first make a logarithm of performance and afterwards divide it by projectsize.

gen logperformance = log(performance+1)

gen logrelperformance = logperformance/projectsize

pwcorr rellogperformance logrelperformance, obs sig

su rellogperformance

* Check the correlation between number of downloads and pageviews, to consides to add pageviews in the analysis.

corr performance T60total_pages

* We see that the two variables are highly correlated. We will take rellogperformance as our indicator for performance.



* Controlvariables:

su STATUSCODE LANG* IA*
tab STATUSCODE
tab STATUSCODE2

* Create a descriptives table

estpost su projectsize logprojectsize degree logdegree brokerage logbrokerage closure_1000 performance relperformance rellogperformance LANG* IA1 IA2 IA3 IA4
```

```
esttab using summm.rtf, cells("count mean sd min max") nomtitle nonumber

* 2. STATISTICAL ANALYSIS

* Check the correlations between the variables

estpost cor projectsize logprojectsize degree logdegree brokerage logbrokerage closure_1000
performance rellogperformance , matrix
esttab using corr.rtf, unstack not noobs compress


* REGRESSION TO PREDICT PERFORMANCE


regress rellogperformance c.logprojectsize##c.logprojectsize
dwstat
rvfplot
estat hettest
vif
est sto m1

regress rellogperformance c.logbrokerage##c.logbrokerage
dwstat
rvfplot
estat hettest
vif
est sto m2

regress rellogperformance c.logprojectsize##c.logprojectsize c.logbrokerage##c.logbrokerage
dwstat
rvfplot
estat hettest
vif
est sto m3
```

```
regress rellogperformance  c.logprojectsize##c.logprojectsize c.logbrokerage##c.logbrokerage
STATUSCODE LANG* IA1 IA2 IA3 IA4
dwstat
rvfplot
estat hettest
vif
est sto m4

esttab m1 m2 m3 m4 using wordH1H2.rtf, se wide r2 noparentheses  noomitted


* REGRESSION TO PREDICT BROKERAGE


* Checking the effect of projectsize on degree

regress degree c.projectsize##c.projectsize
dwstat
rvfplot
estat hettest
vif
est sto m1

regress degree c.projectsize##c.projectsize STATUSCODE LANG* IA1 IA2 IA3 IA4
rvfplot
estat hettest
vif
est sto m2

esttab m1 m2  using wordH3.rtf, se wide r2 noparentheses noomitted


* Checking the effect of degree on brokerage and closure

* Effect of degree on brokerage
```

```
regress brokerage c.degree##c.degree##c.degree
dwstat
rvfplot
estat hettest
vif
est sto m1

regress brokerage c.degree##c.degree##c.degree  STATUSCODE LANG* IA1 IA2 IA3 IA4
dwstat
rvfplot
estat hettest
vif
est sto m2

* effect of degree on closure


regress closure_1000 c.degree##c.degree
dwstat
rvfplot
estat hettest
vif
est sto m3



regress closure_1000 c.degree##c.degree STATUSCODE LANG* IA1 IA2 IA3 IA4
dwstat
rvfplot
estat hettest
vif
est sto m4

esttab m1 m2 m3 m4 using wordH4H5.rtf, se wide r2 noparentheses noomitted
```

```
scatter closure_1000 degree

regress closure_1000 c.degree##c.degree STATUSCODE LANG* IA1 IA2 IA3 IA4 if degree >6
dwstat
rvfplot
estat hettest
vif
est sto m1

esttab m1 using wordhH5special.rtf, se wide r2 noparentheses noomitted

*REGRESSION FOR INTERACTION EFFECT

*H6

regress rellogperformance c.logprojectsize##c.logprojectsize c.logbrokerage##c.logbrokerage
c.closure_1000##c.closure_1000 c.logbrokerage##c.closure_1000
c.closure_1000##c.closure_1000##c.logbrokerage
dwstat
rvfplot
estat hettest
vif
est sto m1


regress rellogperformance c.logprojectsize##c.logprojectsize c.logbrokerage##c.logbrokerage
c.closure_1000##c.closure_1000##c.logbrokerage  STATUSCODE LANG* IA1 IA2 IA3 IA4
dwstat
rvfplot
estat hettest
vif
est sto m2

* Trying again with centered variables

summarize logprojectsize, meanonly
```

```
gen cent_logprojectsize = logprojectsize - r(mean)
su cent_logprojectsize

summarize logbrokerage, meanonly
gen cent_logbrokerage = logbrokerage - r(mean)
su cent_logbrokerage

summarize closure_1000, meanonly
gen cent_closure_1000 = closure_1000 - r(mean)
su cent_closure_1000

regress rellogperformance c.cent_logprojectsize##c.cent_logprojectsize
c.cent_logbrokerage##c.cent_logbrokerage
c.cent_closure_1000##c.cent_closure_1000##c.cent_logbrokerage  STATUSCODE LANG* IA1
IA2 IA3 IA4
dwstat
rvfplot
estat hettest
vif
est sto m3

esttab m1 m2 m3 using wordH6.rtf, se wide r2 noparentheses noomitted



su AGE
gen cent_age2 = AGE - 3305.124 + 0.0004748
su cent_age2

*H7 and H8

regress rellogperformance c.logprojectsize##c.logprojectsize c.logbrokerage##c.logbrokerage
c.closure_1000##c.closure_1000##c.logbrokerage##c.logprojectsize
dwstat
rvfplot
estat hettest
vif
```

```
est sto m1

regress rellogperformance c.logprojectsize##c.logprojectsize c.logbrokerage##c.logbrokerage
c.closure_1000##c.closure_1000##c.logbrokerage##c.logprojectsize STATUSCODE LANG* IA1
IA2 IA3 IA4
dwstat
rvfplot
estat hettest
vif
est sto m2

* The quadratic terms show high a high VIF. This is caused because the original term and the
quadratic term are highly correlated to eachother.
* To mitigate the multicorrilinearity we use the centered variables of the variables which are used
in the quadratic terms:

su logprojectsize logbrokerage closure_1000


regress rellogperformance c.cent_logprojectsize##c.cent_logprojectsize
c.cent_logbrokerage##c.cent_logbrokerage
c.cent_closure_1000##c.cent_closure_1000##c.cent_logbrokerage##c.cent_logprojectsize
STATUSCODE LANG* IA1 IA2 IA3 IA4
dwstat
rvfplot
estat hettest
vif
est sto m3


esttab m1 m2 m3 using wordH7H8.rtf, se wide r2 noparentheses noomitted
```