The main thesis developed and tested in this article is that development stage plays an important moderating role in determining both objective and subjective performance outcomes in free/open source software (F/OSS) projects. This contention is supported by an empirical study of 67 F/OSS projects, with results indicating that subjective performance assessments are calibrated to different objective performance indicators across early and later development stages. The role of team climate variables (trust and shared ideology) in determining both objective and subjective performance also varies across stages. The findings have implications for improving software development processes through appropriate human resource management interventions over the course of F/OSS software projects. Copyright © 2006 John Wiley & Sons, Ltd.

KEY WORDS: free/open source software projects; software project performance; trust; shared ideology; software development stage

1. INTRODUCTION

Free/open source software (F/OSS) has emerged as an important mode of software development and tens of thousands of projects have appeared on sites such as SourceForge.net (http://www.sf.net). While much research attention has been focused on understanding the performance of relatively mature projects such as Linux and Apache (Godfrey and Tu 2000, Mockus et al. 2002), understanding of issues affecting project performance in smaller and less mature F/OSS projects has received less attention. Examining differences in the performance of such projects may be important for understanding how they evolve into larger projects or sustain themselves over time. In this article, we focus on the critical role of the development stage in understanding the variance in F/OSS project performance.

The salience of stages has been established in research on virtual teams as well as in the software engineering literature. Virtual teams have been found to experience distinct stages of team development (Sarker et al. 2001) as they undergo adaptations to technology, organizational or social environments, and team structures (Majchrzak et al. 2000, Qureshi and Vogel 2001). The software engineering literature suggests that client-management and project-management capabilities accumulate in an accretionary manner over time resulting in improved project performance (Whang 1995). Ethiraj et al. (2005) draw upon the dynamic capabilities perspective to suggest that capabilities tend to evolve over time to reflect the joint effects of passive

---

* Correspondence to: Katherine J. Stewart, Department of Decision and Information Technologies, R. H. Smith School of Business, University of Maryland, USA  
† E-mail: kstewart@rhsmith.umd.edu  
Copyright © 2006 John Wiley & Sons, Ltd.
learning-by-doing and deliberate investments in learning and making improvements. Thus, project stage is likely to be a salient contingency factor in understanding F/OSS projects’ performance.

The performance of teams, including virtual teams and software development teams, has often been viewed in terms of input–processing–outcome (IPO) models (Martins et al. 2004, Ilgen et al. 2005). Assessing the ongoing outcomes of software projects is inherently difficult because, for example, the intangible nature of software makes it difficult to obtain accurate estimates of the proportion of work completed, which may in turn promote misperceptions regarding project status (Snow and Keil 2002). To add to the difficulty of measuring performance, software projects are dynamic and tend to have volatile requirements that cause the project scope to change frequently (Johnson 1999). Such issues could be exacerbated in the case of F/OSS projects that may not have clear requirements to begin with (Scacchi 2002). Adding to the complexity of assessing performance in F/OSS, factors typically seen as representing team inputs may, in the F/OSS context, be seen as representing the performance of the project. For example, because projects often rely on voluntary contributions, attracting developers to work on the projects may be seen as an indicator of performance, whereas in most organizational settings, team size, an input, would be considered only as an antecedent to performance.

In addition to the unique aspects of F/OSS projects regarding input and outcomes, the software development processes used in these projects may vary from those used in proprietary software development contexts. In both contexts, software process improvement continues to be an important issue demanding managerial and researcher attention despite significant advances in tools and techniques supporting software development teams (Dalcher and Raffo 2003). Researchers and practitioners have recognized the need to devise principles and practices to help manage human assets as an integral part of software improvement activities. This has resulted in frameworks such as People Capability Maturity Model (P-CMM) that have been shown to significantly impact software development processes (Vakaslihi 1997). Attention to the people management aspects may be even more important for F/OSS teams that are held together by communal ties (Elliott and Scacchi 2003), and therefore we focus in this research on ‘soft’ aspects of development processes including team trust and shared ideology rather than ‘hard’ aspects such as code management tools.

Using the IPO model as an overarching framework, our goal in this study is to build an understanding of (a) how the roles of factors contributing to F/OSS project performance may vary across the stages of F/OSS development and (b) how F/OSS participants’ subjective views of performance may vary across the development stages. The next section discusses F/OSS project performance in light of both inputs and outcomes and briefly reviews some of the key process variables that have been highlighted in prior work on F/OSS development. The section then develops hypotheses regarding objective (i.e. observable input and outcome) indicators of performance on the basis of a discussion of how processes may evolve across the development stages of an F/OSS project. The second goal is addressed by developing hypotheses suggesting how subjective assessments of F/OSS project performance may vary across development stages on the basis of the psychological processes of calibration and attribution. The third section of the article discusses the methodology used to test our hypotheses and summarizes the results of partial least squares (PLS) analysis. In the last section of the article we summarize contributions, limitations, and future research opportunities, as well as discuss the implications of the findings for practice and theory.

2. RESEARCH MODEL AND HYPOTHESES

In IPO models, inputs typically consist of factors such as team size, task, and technology; processes may include factors such as team goal setting, trust, and control; and outcomes often include work quality, effectiveness, and member satisfaction (Martins et al. 2004). The performance of a team may be viewed as a function of any of these factors, though there is perhaps a natural bias to focus on outcomes. However, in F/OSS, it may be equally important to include inputs as indicators of performance. Because much labor in F/OSS is voluntary, an important component of the input to an F/OSS team is attracting developers to work on the project. Thus, the number of developers working on a project may be seen as
an objective input-related indicator of the project’s performance. Outcomes of an F/OSS project have been viewed in terms of the development work performed, including creating new features, fixing bugs, and responding to user requests (Herbsleb and Mockus 2003). We refer to these types of outcomes generally as task completion. Perceived effectiveness refers to a subjective evaluation of how well an F/OSS team has accomplished its goals. Thus, developer team size, task completion, and perceived effectiveness are the three constructs used to represent different dimensions of performance in our research model.

Several processes related to team climate have been shown to be important in motivating efforts in a team and successfully processing team inputs to create viable outcomes. For example, Jarvenpaa and Leidner (1999) demonstrated the importance of trust to global virtual teams, and trust has also been cited as a potentially important factor in F/OSS (Lerner and Tirole 2002, Stewart and Gosain forthcoming). Cultural factors such as subscribing to a shared ideology have also been explored as an aspect of team climate that may be especially relevant in F/OSS (Bergquist and Ljungberg 2001, Bretthauer 2002, Gosain 2003). In general, prior work provides a strong basis for believing that positive team climate factors will have a positive effect on team inputs, team outcomes, and on subjective assessments of team performance. The research model focuses on trust and the extent to which F/OSS members subscribe to a shared ideology as two such factors.

Figure 1. The Research model

A complicating factor in understanding the performance of F/OSS teams is that they are often engaged in ongoing efforts without a defined end goal. Both input and creation of outcomes occur in ongoing cycles as the project proceeds through different stages of development. The generally recognized stages of software development include planning, requirements analysis, design, testing, implementation, and maintenance (Hoffer et al. 2002). On F/OSS websites (e.g. http://www.sf.net), projects are typically categorized as being in stages labeled planning, pre-alpha development, alpha testing, beta testing, production, and mature. Stages prior to the beta testing stage are those during which developers are most focused on designing, building, and basic testing of the intended functionality; during the beta stage the focus is on user trial and testing to uncover and remove bugs encountered during typical usage scenarios; and in later stages the tested software is considered to be stable. The main thesis of this research is that F/OSS team performance, in terms of objective inputs and outcomes, subjective assessments, and the role of team climate, may be better understood by considering the development stage of the project.

The research model (Figure 1) builds on work that has argued for and supported positive impacts of team climate factors including trust and shared ideology to objective and subjective performance. Considering alternate process factors that may link inputs and outcomes and drawing upon attribution theory and the literature on calibration, the model
adds to this work by examining the moderating role of the development stage. Moderating effects are indicated in the model with dashed lines.

3. THE MODERATING EFFECTS OF THE DEVELOPMENT STAGE ON OBJECTIVE PERFORMANCE FACTORS

Team inputs are converted to team outcomes through processes that develop over time. This fact is critical in F/OSS teams because team members may never have worked together in the past, may be separated by geographical distances, and may be unaware of each other’s contextual differences and specific circumstances as the project is initiated. By observing the relationship between member actions and the outcomes that ensue, team members and project managers can select those activities that appear to generate the desired effects most consistently and effectively (Nelson and Winter 1982). Thus, as teams work together for more prolonged periods they develop evolved routines and procedures to facilitate efficient conversion of inputs to outcomes (Ethiraj et al. 2005).

In addition to the learning-from-experience effects, teams are known to adapt to environmental influences as they go through stages of development and maturation over time (Gersick 1988). As an F/OSS project evolves and advances to later stages of development, the coordination and control mechanisms used in the project may become more formalized and ingrained. In this manner, the project should become more effective over time at converting the input of many developers into the outcome of task completion.

H1: The effect of developer team size on task completion will be more positive in later project development stages than in early project development stages.

In the early stages of a project, when formalized procedures and work routines may be lacking, trust and shared ideology may substitute as mechanisms to facilitate progress (Bergquist and Ljungberg 2001, Kirsch et al. 2002). Such aspects of team climate are especially important early in a team’s life because objective accomplishments are less likely to have been produced. A positive team climate in the initial stages will enhance the extent to which team members feel empowered and bolster confidence in the group’s ability to pursue and achieve their goals (Turner 1991). Having clear goals has been shown to correlate to software development team performance (Sonnentag et al. 1994) and may act as a motivating force for developers in later stages. In earlier stages when such goals have yet to become commonly agreed upon, a positive team climate may serve to maintain motivation even in the face of an unclear future for the project. A positive team climate can lead to better mobilization of group members towards directed ends by increasing their level of personal commitment through moral imperatives (Shamir et al. 1993). In the earlier stages, when expectations may not yet be firmly established, teams are most vulnerable to members who may be disruptive. However, a positive team climate tends to isolate such individuals and induce self-doubt in their social position (Turner 1991).

As a project advances to later stages the importance of team climate may be reduced as team members become more directly motivated by the history of accomplishments in the project. Using established work routines, the project administrator may be able to implement meaningful behavior or outcome-based control mechanisms because potential reputational rewards from the project (Hars and Ou 2002) are more consequential.

H2: The effects of trust and shared ideology on task completion will be more positive in early project development stages than in later project development stages.

In addition to having a greater effect on task completion in early stages, positive team climate may be expected to have a stronger effect on attracting and retaining developers in early stages. Developers may associate and remain with a project for many reasons including the utility of the software produced, learning and reputational benefits that may accrue from participation, and psychological benefits derived from membership (Hann et al. 2002, Hars and Ou 2002). The last may be the most important in early stages before the project has established a high level of utility or the recognition sufficient to activate reputational benefits. A positive team climate leads to personal commitment by increasing the salience of collective identities (Shamir et al. 1993). A positive team climate is expected to lead to the construction of a social identity, and in the initial stages will facilitate a process of ‘depersonalization’ such that group members seek to conform to in-group prototypes (Bar-Tal 2000). Elements of a shared ideology such
as shared beliefs have been shown to result in the creation of a social identity, increased group cohesiveness, and differentiation from outgroups (Turner et al. 1987, Hogg et al. 1995). Similarly, a climate of trust will motivate developers to join and continue with a project by reducing potential concerns of opportunistic behavior or incompetence on the part of other members of the team. Thus, attracting and retaining developers in early stages may rely more on having a positive team climate, whereas in later stages having clear goals and a track record may also contribute to developer attraction and retention.

H3: The effects of trust and shared ideology on developer team size will be more positive in early project development stages than in later project development stages.

4. MODERATING EFFECTS OF THE DEVELOPMENT STAGE ON PERCEIVED PERFORMANCE

Understanding subjective views of performance in F/OSS is important because such measures are often used as proxies for objective performance by researchers and also because F/OSS projects rely on the continued efforts of project administrators who are often volunteers. If these individuals perceive the project to be performing well, they are more likely to continue devoting their efforts to it, whereas if they perceive performance as poor they may be more likely to abandon the project. Task completion and developer team size may provide project administrators with objective information to ‘calibrate’ (Alba and Hutchinson 2000) their subjective assessment of team performance. Ideal calibration entails a precise match between subjective assessments of performance and the corresponding objective measures. Work on calibration in psychology (Bjorkman 1994, Gigerenzer et al. 1991) has developed ‘ecological’ models of the calibration of probability judgments. The main premise of these models is that people internalize the associations between cues and events in the world (variously termed ecological validities or external cue validities), and draw upon this internalized knowledge when making judgments (Brenner 2003). Objective indicators of performance such as task completion and developer team size constitute external cues that may enable project administrators to evaluate their success. As the objective indicators of performance increase, they will also lead to an increase in perceptions of team effectiveness. However, the availability and perceived relevance of objective indicators may vary across different project stages.

As a F/OSS project matures and advances to later stages of development, a body of outcomes accumulate by which the project may be judged. In contrast, during the earlier stages of development, the amount of input to the team may be considered a more appropriate objective indicator of performance than outcomes. For this reason, calibration of perceived performance is more likely to be based on developer team size in early stages and task completion in later stages. Developer team size should be more important to perceived performance in early stages because it is a signal that the project will be successful over time. However, in later stages when more information is available on what has actually been produced, then developer team size becomes less important and task completion becomes more relevant.

H4: The effect of developer team size on perceived effectiveness will be more positive in early project development stages than in later project development stages.

H5: The effect of task completion on perceived effectiveness will be more positive in later project development stages than in early project development stages.

In addition to reflecting objective performance, subjective judgments may be influenced by psychological biases. One such bias may result from attribution. Attribution is the process by which people make causal inferences. Weiner (1986) suggests that an important dimension of attribution about others’ behavior concerns the locus of activity: whether behavior is inferred to have been determined internally, by the disposition, effort or ability of the person, or externally, by characteristics of the situation outside the direct control of the person. A fundamental attribution error is the tendency of people to overestimate the importance of internal factors relative to external (situational) factors when making attributions (Ross 1977). When people do not have situational explanations or information, they tend to default to making internal attributions (Jones and Nisbett 1972). Thus in the absence of information to calibrate subjective assessments to objective performance, subjective assessments may be more influenced by attribution errors.
Distributed teams are susceptible to greater possibility of attribution errors, as disconfirming information may be less visible (Cramton 2002). A climate of high trust and shared ideology may encourage positive attributions and reduce the likelihood of negative attributions (Ferrin and Dirks 2003). For example, perceiving team members as caring, capable, and of high integrity (i.e. trustworthy) may affect perceptions of the extent to which team members have followed through on their expected roles (Korsgaard et al. 2002). Over the life of a project, team members have the opportunity to observe others’ behavior, and may therefore be better able to objectively evaluate performance. Thus, the role of team climate in influencing subjective assessments of performance may diminish over time.

H6: The effects of trust and shared ideology on perceived effectiveness will be more positive in early project development stages than in later project development stages.

5. METHODS

We collected data related to F/OSS development projects hosted on SourceForge (www.sourceforge.net). SourceForge provides the open source community with a centralized infrastructure for developers to control and manage F/OSS projects. We chose to sample projects on SourceForge because it is the largest and most well-known online repository of F/OSS projects, because it afforded the opportunity to collect data on objective variables of interest in the study, and because it allowed us to identify project administrators in order to collect data on the subjective variables in the study. In addition to hosting a large segment of the population of F/OSS projects, SourceForge provides these projects with a standard technology toolset, thereby reducing variance in team effectiveness that may be due to differences in technology used to support workflow, code distribution, versioning, etc.

6. DATA COLLECTION

Data was collected using an online survey and from the SourceForge website. We selected projects from two categories on SourceForge – communications (BBS, chat, and ICQ) and multimedia (audio, video, and graphics 3D rendering). We limited the sample to two similar domains to control for possible differences across projects in very different product categories. For instance, an operating system project may involve a different order of complexity than a bulletin board tool project. After selecting categories, we ensured that the projects selected had some activity in the past week in terms of contributions to the code repository; requests for bug-fixes, support, patches or features; or in terms of page views. This was done to ensure that the sample included ongoing projects that had not been abandoned by their developers. Finally, projects were required to have at least two developers to be included in the study because the model is concerned with team processes and dynamics. One hundred and ninety-one projects met all criteria. A subset was randomly selected to pilot-test the survey. For the pilot test, respondents were asked to complete all Likert scale items and answer open-ended questions asking if any of the items were unclear, if they had problems understanding or answering any questions, or if there were ways the survey itself could be improved. Sixteen administrators provided usable responses, and none of them indicated any problems in the survey. Personalized invitations were sent to the remaining project administrators in the sample requesting their participation.

Participants were offered a chance to win a $100 lottery, as well as an opportunity to be informed of the results. Because many individuals are involved in multiple open source projects, the survey instructed the administrators involved with multiple projects to respond with reference to the software project with which they were most involved – this resulted in seven respondents reporting on projects not in the communications or multimedia categories. These seven projects all focused on providing software to support other functional applications; therefore they were categorized as utility applications. Fifty-one administrators provided usable responses to the survey. After analysis of variances (ANOVAs) confirmed that the pilot-test respondents did not differ significantly on any of the outcome measures from the respondents in the second round, the two sets of responses were merged for analysis, resulting in a sample of 67 projects.
7. OPERATIONALIZATION OF CONSTRUCTS

Trust has been conceptualized as a multidimensional construct (McAllister 1995, McKnight and Chervany 2001–2002). We chose to use McAllister’s conceptualization and measures capturing affective and cognitive components of trust because both affective and cognitive factors have been shown to be important in IPO studies of virtual team processes (Martins et al. 2004). The scales were reworded to reflect the change from the original interpersonal level to a team-level assessment (Ammeter 2000). Three items were created to measure the extent to which the group shared an ideology based on prior work that has conceptualized ideology as being composed of norms, beliefs, and values (Trice and Beyer 1993). Perceived effectiveness was operationalized using a five-item scale that reflects how successful the project is perceived to be in meeting its objectives. This measure was similar to earlier measures of software team effectiveness (Zack 1995). All items are included in the Appendix.

Development stages reported on SourceForge at the time of data collection ranged from one for projects in the planning stage to six for projects that had reached a mature state. Intermediate stages were pre-alpha, alpha, beta, and production. Because producing a beta version is a clear and significant milestone in a project, we segregated projects in these stages into three categories: pre-beta, beta, and post-beta. This allowed us to create categories with comparable numbers of projects in each category (22 of the projects were in the pre-beta stage, 18 in beta, and 27 in post-beta).

Developer team size was measured as the number of members associated with the project on SourceForge. Consistent with prior examinations of F/OSS projects, the distribution of team sizes in the sample was slightly skewed, with a mean of 10.39, but both with a median and a mode of 7. Deviation from the normal distribution was relatively minor and the PLS methodology used to test hypotheses was robust to departures from normality; therefore we chose not to transform the variable. The SourceForge site tracks the number of requests for bug-fixes, patches, support, and new features on each project as well as the number of such requests that have an uncompleted (open) status. We calculated task completion as the percentage of tasks completed: [(total requests – requests open)/total requests × 100]. This operationalization is in line with other studies of software teams that use change requests as a measure of software work accomplishment (Herbsleb and Mockus 2003). Because it includes requests for features in addition to requests for changes to existing code, the task completion measure may be relevant even at relatively early stages of project development. However, there were 12 projects that had no task data available; therefore no task completion measure could be calculated (these 12 included projects across all development stages).

All of these measures are likely to be a function of how long the project has been in existence, and therefore project age (number of months since project inception as reported by the administrator) was used as a control variable in the model. Effectiveness may also be influenced by the experience of the project administrator, as more experienced administrators may be able to better manage team efforts. We captured and controlled one indicator of experience by asking administrators how many F/OSS projects they had been involved with (admin. experience). Descriptive statistics are included in Table 1.

8. ANALYSIS

We used ANOVA to assess whether projects varied significantly across categories. We then used PLS to assess measurement properties and test the hypotheses. The PLS methodology enables tests of both the measurement and structural model, allows for small sample sizes, and accommodates possible departures from normal distribution. In order to test hypotheses involving interaction effects, we followed Chin et al.’s (2003) advice by first centering the independent variables and then calculating the cross products using the centered items. Each resulting cross product was then included as an item

Table 1. Descriptive statistics

<table>
<thead>
<tr>
<th></th>
<th>N</th>
<th>Min.</th>
<th>Max.</th>
<th>Mean</th>
<th>Std. deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project age (months)</td>
<td>67</td>
<td>1</td>
<td>108</td>
<td>25.50</td>
<td>21.672</td>
</tr>
<tr>
<td>Developer team size</td>
<td>67</td>
<td>2</td>
<td>48</td>
<td>10.39</td>
<td>9.224</td>
</tr>
<tr>
<td>Admin. experience (projects)</td>
<td>67</td>
<td>1</td>
<td>20</td>
<td>4.19</td>
<td>3.963</td>
</tr>
<tr>
<td>Task completion (%)</td>
<td>55</td>
<td>0</td>
<td>100</td>
<td>53.81</td>
<td>29.729</td>
</tr>
</tbody>
</table>
on the interaction construct. To fully utilize the data and maximize power, we ran separate models for each dependent variable (so that we could include projects with missing data on task completion in the model predicting developer team size).

9. RESULTS

Because data was obtained on projects in three different categories (35 multimedia, 25 communications, and 7 utilities), we conducted ANOVA tests to determine whether the projects were significantly different on variables of interest across categories. These tests did not show a significant difference in the mean values of any of the dependent variables across the categories, and therefore the three sets of projects were pooled for analysis.

We assessed construct validity using the model predicting perceived effectiveness. Confirmatory factor analysis in PLS was conducted to examine construct validity. In the initial analysis, one item for each of the affective trust, shared ideology, and perceived effectiveness scales failed to load on the construct. Three items for the cognitive trust scale failed to load. These items were dropped and the model was re-run. Results then indicated that all items loaded highly on the expected construct with lower cross loadings (Table 2). While the factor analysis results provide evidence of convergent and divergent validity, in the PLS methodology (Agarwal and Karahanna 2000) divergent validity is typically assessed using the average variance extracted (AVE) (i.e. the average variance shared between a construct and its measures). The square root of this measure should be greater than the variance shared between the construct and other constructs in the model (Fornell and Larcker 1981).

---

Table 2. Factor loadings

<table>
<thead>
<tr>
<th></th>
<th>Shared ideology</th>
<th>Perceived effectiveness</th>
<th>Affective trust</th>
<th>Cognitive trust</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ideology 1</td>
<td>0.890</td>
<td>0.135</td>
<td>0.314</td>
<td>0.319</td>
</tr>
<tr>
<td>Ideology 2</td>
<td>0.973</td>
<td>0.152</td>
<td>0.369</td>
<td>0.277</td>
</tr>
<tr>
<td>Effective 1</td>
<td>0.130</td>
<td>0.831</td>
<td>0.532</td>
<td>0.313</td>
</tr>
<tr>
<td>Effective 2</td>
<td>−0.100</td>
<td>0.801</td>
<td>0.187</td>
<td>0.087</td>
</tr>
<tr>
<td>Effective 3</td>
<td>0.102</td>
<td>0.793</td>
<td>0.325</td>
<td>0.267</td>
</tr>
<tr>
<td>Effective 4</td>
<td>0.280</td>
<td>0.635</td>
<td>0.462</td>
<td>0.545</td>
</tr>
<tr>
<td>Affective trust 1</td>
<td>0.397</td>
<td>0.288</td>
<td>0.634</td>
<td>0.265</td>
</tr>
<tr>
<td>Affective trust 2</td>
<td>0.299</td>
<td>0.429</td>
<td>0.835</td>
<td>0.433</td>
</tr>
<tr>
<td>Affective trust 3</td>
<td>0.277</td>
<td>0.470</td>
<td>0.871</td>
<td>0.427</td>
</tr>
<tr>
<td>Affective trust 4</td>
<td>0.243</td>
<td>0.243</td>
<td>0.643</td>
<td>0.539</td>
</tr>
<tr>
<td>Cognitive trust 1</td>
<td>0.080</td>
<td>0.254</td>
<td>0.400</td>
<td>0.758</td>
</tr>
<tr>
<td>Cognitive trust 2</td>
<td>0.186</td>
<td>0.376</td>
<td>0.410</td>
<td>0.827</td>
</tr>
<tr>
<td>Cognitive trust 3</td>
<td>0.458</td>
<td>0.392</td>
<td>0.406</td>
<td>0.802</td>
</tr>
</tbody>
</table>

---

The 7 utility projects were not initially targeted for inclusion, but project administrators reported on these, as they were most closely involved with these projects.
Table 3 shows that this is the case. The table also displays the composite reliability, which is calculated as explained in Agarwal and Karahanna (2000). All scales showed acceptable (>0.7) composite reliability.

The results of the PLS analyses are shown in Table 4. The path coefficients in the model were assessed using the jackknife routine. To examine the nature of the interaction effects, we graphed the predicted value of each dependent variable as a function of each independent variable for each of the three development stages. The example in Figure 2 shows that while the overall slope of developer team size on task completion was negative, it was more positive in later stages than early stages. The significant interaction effects are summarized in Table 5, which describes the results of all hypothesis tests.

10. DISCUSSION AND IMPLICATIONS

The study demonstrated several significant moderating effects of the development stage on performance measures. These results supported the overall thesis underlying the research model, which was that the dynamics of performance in F/OSS projects change as projects move through different stages. However, results of specific hypothesis tests were mixed, supporting some of the hypothesized relationships but also revealing some unexpected findings.

As hypothesized, the outcome measure, task completion, had the most positive effect on perceived effectiveness in later development stages, and the input factor, developer team size, had the most positive effect on perceived effectiveness in earlier stages. These effects support the argument that administrators calibrate their assessments to objective measures but that the relevance of input and outcome measures to calibration changes over time. Developer team size also had a more positive effect on task completion in later stages than in early stages, which was expected; however, the overall effect was negative. While we did not hypothesize the direction of a main effect, intuitively one might expect greater input to lead to better outcomes. The 2Lines in Figure 2 pass through the origin because variables were centered in order to avoid multi-collinearity in the interaction terms.
Table 4. PLS results

<table>
<thead>
<tr>
<th></th>
<th>Developer team size</th>
<th>Task completion</th>
<th>Perceived effectiveness</th>
</tr>
</thead>
<tbody>
<tr>
<td>Controls</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>Project age</td>
<td>0.116&lt;sup&gt;a&lt;/sup&gt;</td>
<td>0.105&lt;sup&gt;a&lt;/sup&gt;</td>
<td>0.080</td>
</tr>
<tr>
<td>Admin. experience</td>
<td>0.018</td>
<td>0.424&lt;sup&gt;c&lt;/sup&gt;</td>
<td>0.159&lt;sup&gt;a&lt;/sup&gt;</td>
</tr>
<tr>
<td>Development stage</td>
<td>0.117&lt;sup&gt;a&lt;/sup&gt;</td>
<td>0.325&lt;sup&gt;b&lt;/sup&gt;</td>
<td>0.155</td>
</tr>
<tr>
<td>Team climate constructs</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>Shared ideology</td>
<td>–0.369&lt;sup&gt;c&lt;/sup&gt;</td>
<td>–0.245&lt;sup&gt;a&lt;/sup&gt;</td>
<td>–0.076</td>
</tr>
<tr>
<td>Shared ideology × development stage</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>Affective trust</td>
<td>0.485&lt;sup&gt;c&lt;/sup&gt;</td>
<td>0.275</td>
<td>0.363&lt;sup&gt;b&lt;/sup&gt;</td>
</tr>
<tr>
<td>Affective trust × development stage</td>
<td>–0.022</td>
<td>–0.092</td>
<td>0.126&lt;sup&gt;b&lt;/sup&gt;</td>
</tr>
<tr>
<td>Cognitive trust</td>
<td>–0.070</td>
<td>0.182&lt;sup&gt;a&lt;/sup&gt;</td>
<td>–0.005</td>
</tr>
<tr>
<td>Cognitive trust × development stage</td>
<td>–0.355&lt;sup&gt;c&lt;/sup&gt;</td>
<td>0.002</td>
<td>–0.050</td>
</tr>
<tr>
<td>Objective performance constructs</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>Developer team size</td>
<td>–</td>
<td>–0.202&lt;sup&gt;a&lt;/sup&gt;</td>
<td>0.139</td>
</tr>
<tr>
<td>Developer team size × development stage</td>
<td>–</td>
<td>0.193&lt;sup&gt;b&lt;/sup&gt;</td>
<td>–0.151&lt;sup&gt;a&lt;/sup&gt;</td>
</tr>
<tr>
<td>Task completion</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>Task completion × development stage</td>
<td>–</td>
<td>–</td>
<td>0.012</td>
</tr>
<tr>
<td>R&lt;sup&gt;2&lt;/sup&gt;</td>
<td>0.377</td>
<td>0.387</td>
<td>0.405</td>
</tr>
</tbody>
</table>

Values in table are path coefficients.
<sup>a</sup> p < 0.10.
<sup>b</sup> p < 0.05.
<sup>c</sup> p < 0.01.
<sup>d</sup> p < 0.001.

Table 5. Results

<table>
<thead>
<tr>
<th>Hypothesis (H)</th>
<th>Outcome</th>
</tr>
</thead>
<tbody>
<tr>
<td>H1: The effect of developer team size on task completion will be more positive in later project development stages than in early project development stages.</td>
<td>Supported</td>
</tr>
<tr>
<td>H2: The effects of trust and shared ideology on task completion will be more positive in early project development stages than in later project development stages.</td>
<td>Contradicted for shared ideology, no effect for trust</td>
</tr>
<tr>
<td>H3: The effects of trust and shared ideology on developer team size will be more positive in early project development stages than in later project development stages.</td>
<td>Supported for cognitive trust, no effect for affective trust, contradicted for shared ideology</td>
</tr>
<tr>
<td>H4: The effect of developer team size on perceived effectiveness will be more positive in early project development stages than in later project development stages.</td>
<td>Supported (marginal effect)</td>
</tr>
<tr>
<td>H5: The effect of task completion on perceived effectiveness will be more positive in later project development stages than in early project development stages.</td>
<td>Supported</td>
</tr>
<tr>
<td>H6: The effects of trust and shared ideology on perceived effectiveness are more positive in early project development stages than in later project development stages.</td>
<td>Supported for shared ideology, contradicted for affective trust (marginal effect), No effect for cognitive trust</td>
</tr>
</tbody>
</table>

It is interesting that team climate variables may have more positive effects on both developer team size and task completion in early opposition to later development stages because the importance of task completion to team effectiveness is more pronounced in earlier stages. The fact that it did not may be because in F/OSS a larger number of developers imply the availability of more labor to work on tasks, but it also means that more tasks are likely to be generated because more people are scrutinizing the project. Our findings may be interpreted as implying that in early stages the efforts of developers in identifying new tasks outstrip their efforts in completing those tasks, but in later stages task identification and completion even out.

We hypothesized that team climate variables would have more positive effects on both developer team size and task completion in early as opposed to later development stages because the importance...
of climate variables would be reduced as other process variables such as work routines became more established in later stages. This was supported for the effect of cognitive trust on developer team size; however, it was contradicted in that shared ideology had a negative overall relationship to both variables and it was more negative in early as opposed to later stages. Reverse causality may partly explain these results. All else equal, larger groups will naturally have greater diversity among members in terms of beliefs, values, and norms. However, over time as the team works together and the project moves to later development stages, members may develop higher levels of shared ideology. For task completion the results could also imply that shared ideology increases participation in identifying tasks but not in completing them, that groups with stronger shared ideology invest more time in attending to the social needs of the members at the expense of task completion, or that team members with shared ideology tend to have more similar skill sets and approaches to problem-solving, which reduce their ability to complete a more diverse set of tasks. Future research is needed to better understand these effects, as the data collected in this study cannot differentiate among these possible explanations.

Finally, the model proposed that the effect of team climate variables on perceived effectiveness would be reduced in later stages as objective information to calibrate judgments accumulated and attribution biases reduced. Results indicated that shared ideology did follow this pattern; however, there was a marginally significant interaction effect in the opposite direction for affective trust, showing that the positive effect of affective trust on perceived effectiveness was stronger in later stages. The mixed results here indicate that aspects of team climate may vary in their impact over time, and future research should utilize more granular measures to better establish the role – or lack thereof – of team climate in biasing subjective judgments of performance.

11. LIMITATIONS AND FUTURE RESEARCH

It is important to point out some of the key limitations of the study. First, the design of the study is cross sectional and as such it cannot assess true causality between variables. There may be important feedback processes that were not captured – i.e. performance in early stages may influence team climate in later stages. Future research employing longitudinal field studies or experiments can help provide a better understanding of the causal dynamics. Second, collecting measures of multiple constructs using a single survey raises the possibility of common methods bias. Analyses indicated sufficient discriminant validity, and drawing measures of objective effectiveness from SourceForge helps to limit this concern; however, it cannot be eliminated. Third, project administrators were used as key informants because they were in a position to report on how the team functions as a whole and to have the most significant impact on the teams’ dynamics (thus their perceptions may be of greater interest than the perceptions of other individual team members). While these respondents were able to tell us about their teams, perceptions may vary across team members, and administrators may tend to have an upwardly biased view of desirable qualities such as trust. Fourth, we chose the measure of task completion partly because it overcomes known problems with other indicators of software project performance such as lines of code (Von Hipple and Von Krogh 2003). However, projects may use the task tracking functions on SourceForge in different ways (e.g. some may record all project agenda items whereas others could be more selective), and this measure did not take into account any specific evaluation of the kind of tasks listed or the quality of task completion. For example, it is possible that teams could report tasks as having been completed without their having been completed in an appropriate manner. Similarly, the measure of team size recorded on SourceForge may not exactly represent true team membership at all times (e.g. developers may make significant contributions without being listed as a developer, or a person could remain listed for some time after participation ceases). We hope that future efforts might seek more discriminating measures of F/OSS teams’ performance.

12. IMPLICATIONS FOR PRACTICE

The results of this study provide practical insights that may be used to improve software development practices not only in F/OSS projects but also in other
contexts involving distributed software development through self-managing teams. First, the results indicated that objective measures of project performance tend to improve over time and with progress in the development stage, while subjective assessments depended to a greater extent on the project administrators’ experience. This pattern of effects in the control variables implies that direct comparisons of project performance be made only between projects that have similar age and are in a similar stage of development. Further, any comparison across project administrators’ subjective judgments of projects should take into account their level of experience.

Second, the results indicate that different dimensions of project performance tend to have different dynamics and differing inter-relationships over time. This suggests that managers should be careful about evaluating the performance of projects using selective indicators. We find, for example, that having a larger developer team may lead to lower task accomplishment in the initial stages but this effect disappears in the later stages. This suggests that using developer team size to assess F/OSS project success may result in accentuated performance on this metric but reduced task accomplishment in the earlier stages. On the basis of this finding, it may be appropriate to provide larger teams with greater up-front support to formalize processes and understand each other’s different contexts. In order to better manage task accomplishment, larger teams may be set up with more hierarchical structures to channel task requests and assign them to team members.

Third, the results show that the effects of team climate variables on project performance differ across project stages. The actual pattern of effects varies for different components of the climate and dimensions of performance. For instance, for perceived effectiveness, the impact of affective trust increases, the impact of shared ideology decreases, and the impact of cognitive trust has no substantive change across the development stages of the project. A practical guideline that follows is to design interventions focusing on specific team climate constructs based on the stage of a project. Thus, in the later stages of a project, trust-building exercises may have greater payoffs in terms of substantive assessments than in the earlier stages. Typically, project teams pay more attention to developer qualifications such as skill sets in staffing a project. Our findings imply that compatibility in terms of ideological perspectives and interpersonal interactions may also be worth considering.

The implications following from the study are in the spirit of software process improvement models such as People CMM (Curtis et al. 1995) that urge attention to human assets in a software organization. While the recommendations of People CMM are at the level of the organization, the implications of this study are more specific and apply at the level of a software project. Future work could organize such fine-grained recommendations into a People–Project CMM model.

13. IMPLICATIONS FOR RESEARCH

The study adds to the literature examining the impact of team factors such as trust on team performance. The findings are especially significant because past research on team factors has often relied on experimental situations rather than a study of intact teams engaged in real-world software development tasks. This study significantly advances understanding of these factors by pointing to the need to account for the development stage of projects when choosing measures of their effectiveness and when incorporating common antecedents such as trust into research models. It indicates the importance of developing an understanding of how the same actions (e.g. trust building) may have different effects depending on the stage of the project. Further, the study indicates that a shared ideology may not always have positive impacts on team performance. While descriptive accounts of F/OSS projects have suggested that developers are motivated by ideological fervor, this study suggests the need for a more nuanced understanding of the varying roles of a shared ideology across projects and across time.

While research streams related to software development and virtual teams have recognized distinct stages in projects and team development, this variable has not received much attention in models of software project success. Our theoretical development envisages that project milestones such as creating a beta release may function as fulcrums whereby process factors important in earlier stages are substituted by other factors and available performance information in later stages. Future research can build on this theoretical model to examine other
Role of Development Stage in Free/Open Source Software Project

factors that may have stage-varying impacts on performance.

14. CONCLUSIONS

This study combined survey data from 67 F/OSS project administrators with data on their projects gathered from the SourceForge website. The key findings indicated that the development stage of a project plays an important moderating role in determining both objective and subjective project performance. For researchers, the study points to the need to account for the development stage of projects when choosing measures of their effectiveness and when incorporating common antecedents such as trust into research models. For F/OSS administrators, the results indicate that different actions may be more or less effective in enhancing project performance depending on the current development stage of the software.

APPENDIX: SURVEY ITEMS (MEASURED USING 7-POINT LIKERT SCALES)

A.1.1. Shared Ideology

Each of the statements below refers to the participants in your project as a group.

1. Members of this group have shared values, beliefs, and norms.
2. Members of this group have shared customs and hold similar views on appropriate rules for behavior.
3. The principles and ideals of members of this group vary widely. [Item dropped]

A.1.2. Affective Trust

Each of the statements below refers to how the participants in your project feel about each other.

1. Members of the team have a sharing relationship with each other. We can freely share our ideas, feelings, and hopes.
2. On this team we can talk freely with each other about difficulties we are having and know that others will want to listen.
3. Members of the team would feel a sense of loss if we could no longer work together.
4. If a member for this group shared problems with other members, they would respond constructively and caringly.
5. Members of the team have made considerable emotional investments in our working relationships. [Item dropped]

A.1.3. Cognitive Trust

Each of the statements below refers to how the participants in your project feel about each other.

1. Members of the team know that everyone on the team approaches their work with professionalism and dedication.
2. Given the track records of the team members, we see no reason to doubt each other’s competence and preparation for a job.
3. Members of the team believe they will be able to rely on other members of the team not to make a job more difficult by careless work.
4. Members of the team are concerned with monitoring each other’s work*. [Item dropped]
5. Members of the team believe that other members should be trusted and respected as coworkers. [Item dropped]
6. Members of the team consider each other to be trustworthy. [Item dropped]

A.1.4. Perceived Effectiveness

1. This project is successful.
2. This project has met its goals so far.
3. Other people find the output of this project useful.
4. This project is active.
5. The quality of the code on this project is high [Item dropped]

ACKNOWLEDGEMENTS

We thank the anonymous reviewers for helpful suggestions on an earlier version of this article. This research was partially supported by the National Science Foundation award IIS-0347376. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily represent the views of the National Science Foundation.
REFERENCES


