A Look on GitHub Projects Attractiveness through Social Inferences

An Empirical Study

Anton P. Ryabchikov
Institute of Doctoral Studies
Lobachevsky State University of Nizhny Novgorod. National Research University
Nizhny Novgorod, Russia
anton.ryabchikov@gmail.com

***Abstract*—The last years have seen a shift of open source software (OSS) development towards social coding forges. Characterized by high transparency of activities and low contribution barriers, they foster competition among projects for the development resources. Projects striving for success in highly competitive environments need a better understanding of what signals facilitate their attractiveness for contributors. Although a number of recent works discusses community members' inferences stimulated by various project-related signals, little has been done on quantitative exploration of the effect of these inferences on OSS project attractiveness. In this paper we bridge this gap by constructing a causal relationship model of project attractiveness and by verifying it through a case study of 2631 projects, extracted from GitHub social coding forge. We argue, based on model analysis, that project "attractiveness" is well represented by the set of visually perceptible project indicators - repository forks, notification subscribers, stargazers and pull requests. The results obtained empirically demonstrate statistically significant positive impact of exposed contribution opportunities (reflected by the number of issues and commit comments), development activity (the size of the development teams and their contributions) and project owner status (the number of followers) on GitHub project attractiveness for contributors. Our findings show that, compared to other factors, inferred contribution opportunities have the most prominent effect on the response construct.**

***Index Terms*—GitHub, Contribution Attractiveness, Path Modeling**

I. Introduction

The adoption of distributed version control systems (DVCS) in software development process and a wide spread of social networks have resulted in a phenomenon of social coding forges. GitHub is an example of the most trending ones [2]. Our paper being under way, GitHub numbered over 19 million projects (or repositories), among which are the well-known jQuery, reddit, curl, Ruby on Rails, node.js, Erlang, CakePHP, Redis and many others. In addition to traditional software-development support (e.g., code hosting, bug tracking), GitHub fosters social interactions. Its members participate in code inspections, follow other members and subscribe to project repositories to watch them. Built on Git technological basis, GitHub introduced a "fork & pull" model permitting members of the community to work independently over source codes in their own forks, the subsequent pulling of changes into the host repository being made by pull request submissions. These features differentiate GitHub from traditional code forges in that they cut down contribution barriers and enable collaboration transparency, resulting in increased competition among projects for the development resources. Projects striving for success need to take care of their attractiveness for contributors. This poses two questions: (a) how do we measure attractiveness, and (b) what makes some projects more attractive than others?

The concept of "attractiveness" (or "popularity", "use", "usefulness", "user interest", "market success", etc.), defined as the ability of an open source software (OSS) project to attract community users to adopt the project software [27, 28], has been extensively discussed in literature in the context of OSS success measures. However, they can not blindly be applied to GitHub projects. Firstly, some measures, such as code downloads or viewings of project web pages [7], are not suitable for GitHub because of DVCS different contribution model (through pull requests) or user ranking mechanisms ("starring"). Secondly, and more important, even more "universal" measures, as the number of developers who have joined the project team [28] or the number of people in the project mailing lists [6, 7, 27], do not take into account transparent working environment and collaboration awareness, offered by GitHub. As pointed out by Dabbish et al. [8] and Tsay et al. [29], these features gave rise to a broad range of inferences, crucial for participation decision-making. Hence, the concept of "attractiveness" should be revised in light of integrated social features and lower contribution barriers.

From the community member perspective selecting a project for contribution resembles consumer's behavior in a sense of assessment of perceived product characteristics. It has been shown [3] that products consist of a set of cues, which shape various consumer's impressions. The above-stated suggests the possibility to qualify a GitHub project by a set of cues that developers pay attention to in their assessment and further quantify these cues in order to understand their influence on project contribution attractiveness. In this regard, cue utilization theory, successfully employed in marketing, management and recently in OSS analyses [19], serves as a conceptual framework for our study. We build a theoretical causal model of GitHub project attractiveness on the basis of inferences made by GitHub community members, evaluating perceptible project cues. This model helps us answer our research question.

RQ: What is the effect of social inferences on GitHub project popularity in terms of its attractiveness for contributors?

Markus et al. [18] suggested a critical role of attracting contributors on an on-going basis in maintaining project sustainability. Thus, understanding of the influence of signals, OSS projects generate, sheds light on the cornerstone for OSS communities (with voluntary participation) question on the driving forces behind contributors attraction. Despite Tsay et al. [29] posed similar question in their research of social media effects in transparent work environments, their findings were based on interviews and not validated by empirical data. The distinctive feature of our study is the quantification and validation of these effects on a carefully constructed dataset from GitHub, the most widely used DVCS and social coding network. Our research is further discriminated from the previously conducted ones in the following aspects:

- is grounded on inferences made by community members instead of project's raw data metrics (the latter is prevalent in studies of SourceForge or large-scale OSS projects such as Debian, Apache or Mozilla with limited transparency and rigid contribution standards);
- treats "attractiveness" as multidimensional construct, comprised of various levels of user concernment.

The remainder of the paper is organized as follows. In Section II we construct the model and the dataset to answer our research question. Section III presents the approach and the analysis results, while Section IV discusses our findings in light of related works. Section V discloses the validity threats. Finally, Section VI draws conclusions and some ideas for future work.

II. Study Design

A. Research Model and Hypotheses

We build our research model on top of the key conjecture of Midha and Palvia's [19] work on the influence of project's "technical success", defined as the level of efforts expended by developers of the project, on its "market success", a measure of project popularity, defined as the level of interest displayed in the project by its consumers. Since Midha and Palvia have considered SourceForge hosting of OSS projects in their study, we first need to adapt their view of technical and market success to the previously outlined characteristics of GitHub by choosing signals, GitHub members take into account when reasoning about developers' efforts and repository attractiveness. Then, based on the literature review, we formulate two additional hypotheses on the factors affecting contribution attractiveness of GitHub projects.

Our view on project attractiveness is wider than Midha and Palvia's notion of "market success". We regard "attractiveness" as multivariate construct that needs to be assessed from multiple perspectives, which corresponds to Crowston's [7] treatment of OSS success as a multidimensional phenomenon. GitHub project information coming in a form of the number of *forks* and *watchers* has been attributed by Dabbish et al. [8] to both project usefulness and a motivation to contribute, making these signals a reliable manifestation of project attractiveness for contributors. More to that, Lee et al. [15] concludes that *pull requests* can be thought of as an indicator of project popularity, since they induce follower activity, especially if submitted by community celebrities. Lastly, in 2012 GitHub changed how its watching system worked[1]. Instead of users just "watching" a project and subscribing to its events, they currently can either "star" a project to bookmark it and "watch" a project to receive notifications. As Dabbish et al. [8] did not account for that change, we have to include both *watchers* (*subscribers* in current GitHub terminology) and *stargazers* into the model. It should be noted, that two of these quantities - subscribers and stargazers - may more likely be treated as signs of passive interest in the project, whereas forks and pull requests, as per "fork & pull" model, should be attributed to planned or already offered contributions from the community members. Thereby, in this study project "attractiveness" encompasses four complementary levels of community interest in the project - from simple bookmarking to code snippets proposals.

Midha and Palvia's notion of "technical success" is concerned with commit activity of the project team. In GitHub contributors and commits statistics are readily available for every repository. According to Dabbish et al. [8], the amount of commits serves as one of the most vital signals of developers commitment and their level of investment in the project, which is in line with Midha and Palvia's definition of "technical success". To account for the project size, we suggest both the number of developers involved in a project and their commit counts as signals of development activity.

Taking that into account, we reformulate Midha and Palvia's conjecture in the following way:

H1. Development activity positively affects contribution attractiveness.

Von Krogh et al. [14] found that in open source software potential contributors to a project's mailing list often spend a considerable amount of time observing the project mailing list before contributing. This helps them get acquainted with the existing project participants and the culture of the project. In GitHub communications are code-centric, taking place around commits, issues or pull requests. Dabbish et al. [8] reported that registered issues and comments on commits are regarded as opportunities to contribute to the project (not necessarily to the codebase), resulting in our second hypothesis:

[1] https://github.com/blog/1204-notifications-stars

H2. Exposed contribution opportunities have a positive impact on contribution attractiveness.

The influence of SourceForge project leader's characteristics on its popularity as well as on developer's motivation to contribute was studied by Wu and Goh [33] in view of a social network perspective and by Li et al. [16] in light of leadership and motivation theories. These studies showed that leaders' active management style is positively related to the developers' motivation. Likewise, higher leadership centrality is associated with increased level of project popularity. Social network capabilities of GitHub allow its members to "follow" developers. Developers with many followers, i.e., high in-degree, earned a reputation of local celebrities (so-called "rockstars"). In the recent analysis of GitHub "rockstar" developers Lee et al. [15] found a pronounced effect of such members activity in the project on their followers activity. Similarly, in a set of semi-structured interviews of developers Tsay et al. [29] found that GitHub members make use of the following relation and user reputation to make decisions what projects they should attend to. This leads to our third hypothesis:

H3. Project owner status is positively associated with contribution attractiveness.

The resulted research model, incorporating our hypotheses, is presented in Fig. 1, where each inference is manifested by corresponding visual cues. The next section of the paper is devoted to quantitative verification of the model, which would lead us to the understanding of the influence of social inferences on project attractiveness in GitHub context.

B. Dataset Construction

Figure 2 shows that our dataset construction approach is broken down into three steps: (1) extract events from Githubarchive.org archives, (2) query GitHub API for the detailed repositories' data, and (3) calculate repositories' metrics. The application of filtering criteria in the sampling phase is explained below.

At the first step we extracted and parsed adjacent logs of events fetched from Githubarchive.org during a continuous time interval (from March 24, 2014 to March 26, 2014) to assemble an initial sample of repositories. This approach is advantageous in a sense of perfect selection randomness, whereas the fact of repository presence in Githubarchive.org log files indicates its recent activity. Each event encountered in logs was tested against the following criteria.

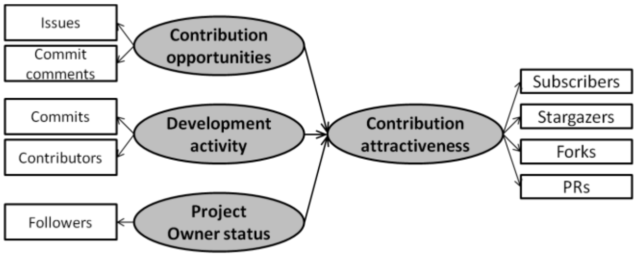


Fig. 1. Research model of the study. Drawing conventions: ellipses represent latent variables; manifest variables are in boxes; the arrows linking latent variables indicate the dependence flow which specifies the path model.

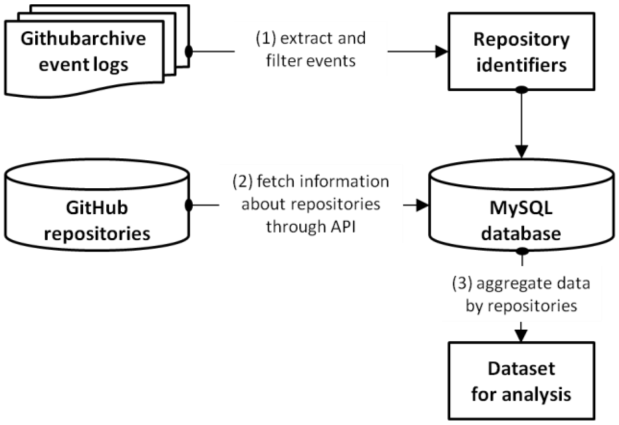

Githubarchive event logs
(1) extract and filter events
Repository identifiers
GitHub repositories
(2) fetch information about repositories through API
MySQL database
(3) aggregate data by repositories
Dataset for analysis

Fig. 2. Overview of our dataset construction approach.

- *An event is linked to a pull request (type "PullRequestEvent").* This can be thought of as screening of projects that make use of "fork & pull" model.
- *The initiating repository of a pull request differs from the target one.* Cross-branch pull requests serve primarily as means of communication.
- *The repository the pull request is destined for is associated with personal account (type "User").* Despite user/organization separation is purely a convenience mechanism on GitHub, the particular problem with organization-owned repositories here is that GitHub does not allow to "follow" organizations. One of the options is to construct an artificial measure of organization status, but this would lead to inconsistency of indicators within a common dataset and introduce contradiction to the notion of "perceptible cue" indicator in our model. Moreover, many projects managed by organizations are in fact associated with personal accounts. At a later date they may be moved to the main organization (this is what happened, e.g., with Storm). We decided to limit our sample by personal repositories.
- *The repository is not a fork.* Main repositories serve as attraction centers for contributors. Herein the contribution itself is assumed to happen by means of "fork & pull" mechanism. In addition to that, the vast majority of forks originate directly from the main repository, not from the other forks.
- *The repository issue tracking feature is enabled (has_issues property equals "true").* Many projects use separate from GitHub defect tracking systems and disallow issues registration on project page [13]. This case should be distinguishable from the absence of issues.

At the second step we wrote a custom Perl script to fetch properties of interest directly from GitHub API for each of the 2729 unique repositories selected from the event logs. The decoded JSON responses then were imported into local MySQL database.

At the third step by means of SQL we calculated model variables listed in Table I. To ensure only active projects in the sample, we followed recommendations of Kalliamvakou et al. [13] and dropped out repositories with no commits in a 12 months time span prior to sampling period. This resulted in a slight 3.6% sample reduction, as expected. The descriptive statistics of the final dataset consisting of 2631 projects, our findings are based upon, is presented in Table I.

III. Data Analysis and Results

Given our study design, there exist several model evaluation approaches, including Multiple Factor Analysis, Structural Equation Modeling, Partial Least Squares Path Modeling (PLS-PM) and others. We preferred PLS-PM for several reasons.

Firstly, PLS-PM has proved to provide better evidence for the existence of relationship in settings where the relationship is sought among hypothetical (latent) constructs, manifested by perceptible indicators (e.g., [32]). Secondly, our evaluation of the research model is more exploratory in nature rather than confirmatory. Lastly, PLS-PM does not impose distributional assumptions on the variables, which is advantageous due to their high skewness (see Table I).

We performed data analysis in accordance with a two-step approach [5]. The first step was the assessment of the measurement model, which is focused on the reliability and validity of the construct measures used. The second step was to test the structural model. R package *plspm* [22] and the bootstrap resampling procedure were used to assess the measurement and structural PLS models.

A. Measurement Model Assessment

At the first step of the data analysis we tested the reliability and validity of the measurement model (Table II). A classical index in reliability analysis is *Cronbach's alpha*, which provides an estimate for the reliability based on the indicator intercorrelations. Hair et al. [11] suggested that a generally accepted lower limit for Cronbach's alpha is 0.7, although that may decrease to 0.6 in exploratory studies. In our case Cronbach's alpha exceeded 0.6 for all constructs. We additionally reported on *Dillon-Goldstein's rho* index, also known as composite reliability, which is considered to be a better indicator than Cronbach's alpha [4], as it is based on the results from the model (i.e., the loadings) rather than the correlations observed between the manifest variables in the dataset. Compared to Dillon-Goldstein's rho, Cronbach's alpha

TABLE I. Selected projects' indicators and descriptive statistics (histograms are in log scale)

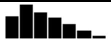
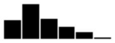
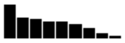
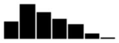
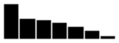
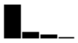
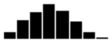
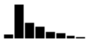
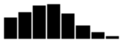
Indicator	Description	min	25%	median	mean	75%	max	Histogram
Forks	Number of repository forks	0.00	2.00	10.00	88.46	45.00	10460.00	
Subscribers	Number of repository subscribers (watchers)	0.00	2.00	5.00	29.42	20.00	1501.00	
Stargazers	Number of repository stargazers	0.00	2.00	19.00	373.30	155.50	29832.00	
PRs	Number of pull requests (excluding intra-branch ones)	0.00	3.00	10.00	52.16	38.00	3764.00	
Issues	Number of non-pull request issues in the repository	0.00	1.00	8.00	68.36	42.50	3924.00	
Comments	Number of commit comments	0.00	0.00	0.00	11.35	4.00	2978.00	
Commits	Total number of commits made by all project contributors	1.00	29.00	90.00	375.10	297.00	31004.00	
Contributors	Number of people contributed to the project	1.00	2.00	4.00	15.57	12.00	1654.00	
Followers	Number of the repository owner followers	0.00	5.00	22.00	220.60	83.00	17310.00	

TABLE II. Results of reliability and validity tests with corresponding 95% confidence intervals (C.I.) for loadings built by means of 300000 bootstrap samples

Construct	Indicator	Factor Loading with 95% C.I.	Cronbach's alpha	Dillon-Goldstein's rho	AVE
Contribution Opportunities	Issues	0.940 [0.920;0.954]	0.776	0.899	0.812
	Comments	0.860 [0.806;0.927]			
Development Activity	Commits	0.770 [0.691;0.863]	0.629	0.844	0.720
	Contributors	0.920 [0.886;0.957]			
Owner Status	Followers	1.000 [1.000;1.000]	1.000	1.000	1.000
Contribution Attractiveness	Subscribers	0.893 [0.858;0.932]	0.870	0.914	0.721
	Stargazers	0.900 [0.862;0.938]			
	Forks	0.873 [0.763;0.955]			
	PRs	0.718 [0.667;0.780]			

provides a lower bound estimate of reliability. The advised lower cut-off value for Dillon-Goldstein's rho is 0.7 [30], whereas in our study 0.84 was the minimum value of the index.

Following the study of Fornell and Larcker [10], we reported average variance extracted (AVE), a criterion of convergent validity, which signifies that a set of indicators represents one and the same underlying construct, i.e., demonstrates its unidimensionality. The AVE for each construct should exceed 0.5 [10], meaning that a latent variable is able to explain more than half of the variance of its indicators on average. Falk and Miller [9] suggested that the factor loading of each indicator should be greater than 0.55. As shown in Table II, the loadings for all our indicators were statistically significant at a 5% confidence level and greater than 0.55. The AVE extracted for all constructs exceeded 0.5.

Next, we applied Fornell-Larcker criterion of discriminant validity by comparing the square root of the AVE for each construct with the correlations among constructs [5]. If a certain construct is more correlated with another construct than with its own measures, there is a possibility that the two constructs are not conceptually distinct. As shown in Table III, the square root of the AVE for each construct (the diagonal line) exceeded correlations between that construct and other constructs, i.e., the joint sets of indicators are not unidimentional.

Thus, the results reported in this section allow us to claim the reliability and validity of the measures used to represent our constructs.

B. Structural Model Assessment

Reliable and valid measurement model estimations permit us to evaluate the structural model. We evaluated the quality of the structural model by examining three indices: the *coefficient of determination (R^2)*, the *redundancy* index and the *Goodness-of-Fit (GoF)* index [30].

The coefficient of determination of the endogenous latent variable (i.e., Contribution Attractiveness) indicates the amount of variance in the construct in question that is explained by its independent latent variables. It is interpreted similarly to multiple regression analysis. Chin [4] describes R^2 values of 0.67, 0.33, and 0.19 in PLS path models as substantial, moderate, and weak, respectively. Our model, explaining 61.3% of the variance in Contribution Attractiveness with 95% bootstrap confidence interval of [55.9; 68.8], formally fell under moderate category. In settings where endogenous latent variable is explained by only a few exogenous latent variables moderate R^2 are considered to be acceptable [12].

The redundancy index serves as a global quality measure of the structural model. It signals how well the independent latent variables predict values of the indicators' endogenous construct. In our model the redundancy index computed for Contribution Attractiveness shows that three independent latent variables predict 44.2% of the variability of Contribution Attractiveness indicators.

The Goodness-of-Fit index has been developed in order to take into account the model performance at both the measurement and the structural level and thus provide a single measure for the overall prediction performance of the model. This index is bounded between 0 and 1, but there is no guidance about what value could be considered an acceptable GoF threshold. In our study the GoF value of 0.68 could be interpreted as if the prediction power of the model is 68%.

TABLE III. Correlations between latent variables (the square root of the construct's AVE are shown in the diagonal)

	Contribution Opportunities	Development Activity	Owner Status	Contribution Attractiveness
Contribution Opportunities	**0.901**	0.501	0.172	0.702
Development Activity	0.501	**0.849**	0.145	0.617
Owner Status	0.172	0.145	**1.000**	0.303
Contribution Attractiveness	0.702	0.617	0.303	**0.849**

The individual path coefficients of the PLS structural model can be interpreted as standardized beta coefficients of ordinary least squares regression. Structural paths, which are statistically significant and whose sign is in line with the direction of theoretically assumed relationships, support a priori formed hypotheses. Path coefficients along with 95% confidence intervals, determined by bootstrapping technique, as well as corresponding hypotheses tests are reported in Table IV. Bootstrap intervals for all path coefficients clearly excluded 0, which proved their significance at a 5% confidence level.

Hence, development activity (H1), exposed contribution opportunities (H2) and project owner status (H3) had significant influences on GitHub repository attractiveness, explaining 61.3% of the variance in dependent latent variable.

IV. Discussion and Related Research

Unprecedented transparency of public projects hosted on GitHub allows community members for a rich set of inferences. Dabbish et al. [8] linked these inferences to perceptible project cues. Tsay et al. [29] further developed this idea by speculating about possible user actions in response to

TABLE IV. Path coefficients and hypotheses

Structural Path	**Path Coefficients and 95% C.I.**	**Conclusion**
H1. Development Activity→ Contribution Attractiveness	0.341 [0.241; 0.458]	Supported
H2. Contribution Opportunities→ Contribution Attractiveness	0.502 [0.373; 0.597]	Supported
H3. Owner Status→ Contribution Attractiveness	0.167 [0.107; 0.232]	Supported

signals broadcasted by an open source ecosystem projects, however, empirical evaluation of their findings was left for future research. Inspired by these works, we validated our hypotheses through the quantitative analysis of a large sample of projects, extracted from GitHub. Having confirmed the hypotheses our model is comprised of, we were able to conclude the positive effect of three types of social inferences, made by GitHub members, on project contribution attractiveness.

We discussed the key aspects of the analysis results as well as their practical implications.

A. The Notion of "Attractiveness"

Various researchers employed the notion of software project "attractiveness" within different contexts. In studies devoted to OSS success projects attractiveness has been expressed in terms of "popularity" [25, 33], "user interest" [20, 27], "use" [17], "signal of market success" [19], these a few being mentioned, with a variety of proxy measures. This fact itself justifies our treatment of attractiveness as multidimensional construct. In a recent study of documentation evolution of GitHub projects Aggarwal et al. [1] defined project popularity as a pure sum of stars, forks and pull requests squared. Their study, however, lacks unidimensionality analysis of the set of indicators, i.e., whether they represent one and the same underlying construct. While we chose a similar set of indicators to represent attractiveness, its unidimensionality was confirmed at the measurement model level (see indices in Tables II-III). Despite one may argue over the conceptual appropriateness of advised construct's indicators, they were homogenous enough to be considered as at least an initial effort in redefining "attractiveness" for highly transparent environments of social coding forges.

B. The Value of Inferences

We found that the project development activity is positively perceived by community members and attracts attention to that project in a form of subscribers, stars and code proposals. Different forms of relationships (direct, lagged, reverse, etc.) between development efforts and project popularity have been previously tested in OSS literature on various SourceForge samples, resulting in somewhat conflicting findings. For example, Midha and Palvia [19] have not found that OSS projects with higher level of activity are more popular, where the levels of activity and popularity are measured by the amount of commits and downloads, respectively. However, the reverse relationship has been tested and confirmed by Stewart et al. [26], suggesting that project popularity may impact the developer activity. By applying similar measures (number of developers and total community size as input, downloads and project page views as output) Crowston et al. [7] have not found significant correlation between project popularity and the number of developers. Finally, Sen et al. [23] studied the determinants of OSS success as measured by the number of subscribers and developers working on an OSS project and found positive and significant impact of both developers on subscribers and subscribers on developers. Our finding concerning the impact of GitHub project development activity on its attractiveness is promising as it suggests that in highly competitive and transparent environments the composite effect of development team size and the amount of contributions made is more prominent than in traditional OSS development.

As GitHub shares many characteristics with social networks, this permits to make cross-comparison. Romero et al. [21] demonstrated that Twitter users' high popularity, in a sense of both obtaining attention and overcoming passivity, does not necessarily imply high influence and vice-versa. Nevertheless, Walther et al. [31] found that Facebook users with an above-average number of friends are associated with higher ratings of both social attractiveness and extraversion. This is in line with Tsay's [29] conjecture on using high status users as a way to determine potentially interesting projects in GitHub. Our conclusion on the positive effect of project owner status onto project contribution attractiveness also favors it.

The finding of the pronounced dominating effect of contribution opportunities, compared to other factors, on the response construct is worth mentioning. In this study we manifested contribution opportunities through project issues (bugs, required features, tasks to work on, etc.), which pinpoint the areas where contributions are welcome, and comments on commits, which often take a form of a discussion on the style, correctness or efficiency of a particular contribution. Steinmacher et al. [24] reported in their review that locating the appropriate task/issue to work on within a project as well as timeliness and relevance of the feedback received from the community were considered to be crucial barriers faced by OSS projects' newcomers. Von Krogh et al. [14], for example, found that in only 16.7% of the cases new project participants were given specific assignments. Thus, in the absence of explicit task assignments project issues are viewed as tasks to choose from. Although GitHub notably cuts down participation barriers by its "fork & pull" mechanism, which practically does not preclude anyone from working on the codebase of a public project, our empirical findings on the significant influence of contribution opportunities on contribution attractiveness suggested that the importance of the mentioned barriers should not be underestimated.

C. Practical Implications

From a practical perspective, the results obtained provide indicators project collaborators should keep track of in order to sustain attractiveness of their repositories in a highly competitive environment. In addition to algebraic signs and statistical significance, differences in magnitude of our research model path coefficients can be interpreted as to which factors project stakeholders should concentrate their attention. Efforts invested into issues management and facilitation of discussions over code snippets will most likely be rewarded by community members, compared to self-reputation management or, rather surprisingly, project development performance. It is worth noting, that in Dabbish et al. [8] inferences on contribution opportunities were expressed by a minority of respondents. Our results pointed to a dominating impact of such inferences. By contrast, project owner status-related inferences turned to be the least influential in our study.

V. Validity Threats

Threats to validity of our study are discussed in accordance with common guidelines [34].

Threats to *conclusion validity* concern the relationship between the predictor and the response constructs. These threats are believed to be mitigated by applying an established two-step model assessment approach accompanied by bootstrap validation.

Threats to *internal validity* relate to the inclusion of repositories to the sample and analysis methods employed. GitHub is considered to have a proportion of projects, unrelated to software development (e.g., toy, educational projects or even file storages) [13]. We believe the criteria we filtered against the initial set of repositories in Section II-B mitigate this threat. For additional criteria of irrelevant projects exclusion one can reference Black Duck Open Hub[2] (former Ohloh) listing of software development projects. The other threats to internal validity are possible biases in manifest variables due to, for example, bots or spam followers [29]. In a future study we plan to implement some heuristics in a data screening procedure.

Threats to *external validity* refer to the possibility to generalize the results obtained herein on OSS communities beyond the scope of GitHub hosting. Although we tested our hypotheses on a fairly large population of projects and the model assessment results proved it to be adequate, it is unclear whether the model also performs well for other forges. While this study is focused solely on GitHub projects (not even on any GitHub project, as per our data screening approach), validation on data from different sources (e.g., Bitbucket, Black Duck Open Hub) is appreciated.

Threats to *reliability validity* concern the reproducibility of this study. The set of projects we analyzed was made publicly available at the GitHub repository[3]. We also attempted to reveal all the details of the analysis stage to make our study reproducible.

VI. Conclusions and Future Work

In this paper through a case study of the repositories from GitHub social coding forge we studied the impact of inferences, made by its community members on the basis of perceived project cues, on project attractiveness for contributors. Despite a number of recent works linked project-generated signals to users' conclusions about, e.g., developer skills, commitment, project or artifact importance, the effect we questioned about has got little attention. To evaluate the effect of inferences, we built a causal relationship model of project attractiveness and analyzed it by means of PLS path modeling technique on a dataset of 2631 GitHub repositories.

We instantiated project attractiveness by a set of visually perceptible indicators – the number of forks, project notification subscribers, stargazers and pull requests. Measurement model analysis proved them to be a reliable reflection of project attractiveness. This finding showed how project attractiveness could be defined in the case of highly transparent environments of social coding forges. Furthermore, we tested our hypotheses seeking to understand how the development activity, contribution opportunities and project owner status influence its attractiveness. Structural model analysis indicated positive and significant effect of these inferences. Compared to development activity and project owner status, contribution opportunities, signaled by issue listings and comments on commits, have the dominating effect on the response construct. This signifies, in agreement with the previous research, that the exposure of issues to work on and proper community feedback are essential factors in the process of newcomers' onboarding in OSS projects.

We believe our findings offer strong empirical evidence to support the value of social inferences in transparent collaborative environments. Practically, the study provides guidelines for OSS project stakeholders on how to measure and manage the level of interest displayed in the project by community users to ensure a sustainable stream of contributions. Following these guidelines could favor the overall improvement of success rates of OSS projects.

However, many questions referring to the driving forces behind project attractiveness remain unanswered. For example, how can our research model be extended to incorporate other relevant inferences people make when transparency is integrated into the forge? How do various moderating factors (e.g., recency, frequency and location of commits or project maturity stage) influence the strength and direction of the relationship between independent and dependent constructs? How can the concept of "attractiveness" be further refined? Another potentially interesting aspect of analysis deals with unobserved heterogeneity in sampled projects, where looking for local models characterized by class-specific model parameters may provide new insights into the problem treatment. These and allied questions lay down directions for future research.

References

[1] K. Aggarwal, A. Hindle, and E. Stroulia, "Co-evolution of project documentation and popularity within Github," In Proceedings of the 11th Working Conference on Mining Software Repositories (MSR'14), 2014, pp. 360–363.

[2] D. Berkholz, "The size of open-source communities and its impact upon activity, licensing, and hosting," at http://redmonk.com/dberkholz/2013/04/22/the-size-of-open-source-communities-and-its-impact-upon-activity-licensing-and-hosting, accessed 20 August 2014.

[3] R. Burnkrant, "Cue utilization in product perception," Advances in consumer research, vol. 5(1), 1978, pp. 724-730.

[4] W. W. Chin, "The partial least squares approach for structural equation modeling," in Modern methods for business research, G. A. Marcoulides, Ed. London: Lawrence Erlbaum Associates, 1998, pp. 295–336.

[5] W. W. Chin, "How to write up and report PLS analyses," in Handbook of partial least squares, V. Esposito Vinzi, W. W. Chin, J. Henseler, and H. Wang, Eds. Springer Berlin Heidelberg, 2010, pp. 655-690.

[2] www.openhub.net

[3] github.com/aryabchi/ggha

[6] K. Crowston, H. Annabi, and J. Howison, “Defining open source software project success,” in Proceedings of the 24th International Conference on Information Systems (ICIS 2003), Seattle, WA, 2003.

[7] K. Crowston, J. Howison, and H. Annabi, “Information systems success in free and open source software development: Theory and measures,” Software Process: Improvement and Practice, vol. 11(2), 2006, pp. 123–148.

[8] L. Dabbish, C. Stuart, J. Tsay, and J. Herbsleb, “Social coding in GitHub: transparency and collaboration in an open software repository,” In Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work (CSCW’12), 2012, pp. 1277-1286.

[9] R. Falk and N. Miller, A prime for soft modeling. Ohio: University of Akron Press; 1992.

[10] C. Fornell and D. Larcker, “Evaluating structural equation models with unobservable variables and measurement error,” Journal of Marketing Research, vol. 18(1), 1981, pp. 39–50.

[11] J. Hair, R. Anderson, R. Tatham, and W. Black, Multivariate data analysis, 5th ed., Englewood Cliffs, NJ: Prentice-Hall, 1998.

[12] J. Henseler, C.M. Ringle, and R.R. Sinkovics, “The use of partial least squares path modeling in international marketing,” Advances in International Marketing, vol. 20, 2009, pp. 277-319.

[13] E. Kalliamvakou, G. Gousios, K. Blincoe, L. Singer, D. M. German, and D. Damian, “The promises and perils of mining GitHub,” In Proceedings of the 11th Working Conference on Mining Software Repositories (MSR’14), 2014, pp. 92-101.

[14] G. von Krogh, S. Spaeth, and K. R. Lakhani, “Community, joining, and specialization in open source software innovation: a case study,” Research Policy, vol. 32, no. 7, 2003, pp. 1217–1241.

[15] M. J. Lee, B. Ferwerda, J. Choi, J. Hahn, J. Y. Moon, and J. Kim, “Github developers use rockstars to overcome overflow of news,” In CHI’13 Extended Abstracts on Human Factors in Computing Systems, ACM, 2013, pp. 133-138.

[16] Y. Li, C. H. Tan, and H. H. Teo, “Leadership characteristics and developers’ motivation in open source software development,” Information & Management, vol. 49(5), 2012, pp. 257-267.

[17] J. Long, “Understanding the role of core developers in open source software development,” Journal of Information, Information Technology, and Organizations, vol. 1, 2006, pp. 75–85.

[18] M. Markus, B. Manville, and C. Agres, “What makes a virtual organization work?,” Sloan Management Review, vol. 42(1), 2000, pp. 13–26.

[19] V. Midha and P. Palvia, “Factors affecting the success of open source software,” Journal of Systems and Software, vol. 85(4), 2012, pp. 895–905.

[20] E. Petrinja and G. Succi, “Two evolution indicators for FOSS projects,” In Proceedings of the 8th IFIP WG 2.13 International Conference, 2012, pp. 216-232.

[21] D. M. Romero, W. Galuba, S. Asur, and B. A. Huberman, “Influence and passivity in social media,” In Machine learning and knowledge discovery in databases, vol.6913, D. Gunopulos et al., Eds. Springer Berlin Heidelberg, 2011, pp. 18-33.

[22] G. Sanchez, L. Trinchera, and G. Russolillo, “plspm: tools for partial least squares path modeling (PLS-PM),” R package version 0.4.1, 2013.

[23] R. Sen, S. S. Singh, and S. Borle, “Open source software success: Measures and analysis,” Decision Support Systems, vol. 52(2), 2012, pp. 364-372.

[24] I. Steinmacher, M. A. G. Silva, and M. A. Gerosa, “Barriers faced by newcomers to open source projects: a systematic review,” In Open Source Software: Mobile Open Source Technologies, vol. 427, L. Corral et al., Eds. Springer Berlin Heidelberg, 2014, pp. 153-163.

[25] K. Stewart and T. Ammeter, “An exploratory study of factors influencing the level of vitality and popularity of open source projects,” In Proceedings of the 23rd International Conference on Information Systems, 2002, pp. 853–857.

[26] K. Stewart, A. Ammeter, and L. Maruping, “A preliminary analysis of the influences of licensing and organizational sponsorship on success in open source projects,” In Proceedings of the 38th Annual Hawaii International Conference on System Sciences (HICSS’05), 2005, p. 197.

[27] K. Stewart, A. Ammeter, and L. Maruping, “Impacts of license choice and organizational sponsorship on user interest and development activity in open source software projects,” Information Systems Research, vol. 17(2), 2006, pp. 126–144.

[28] C. Subramaniam, R. Sen, and M. Nelson, “Determinants of open source software project success: A longitudinal study,” Decision Support Systems, vol. 46(2), 2009, pp. 576–585.

[29] J. Tsay, L. Dabbish, and J. Herbsleb, “Social media in transparent work environments,” In Proceedings of the 6th International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE), IEEE, 2013, pp. 65-72.

[30] V. E. Vinzi, L. Trinchera, and S. Amato, “PLS path modeling: from foundations to recent developments and open issues for model assessment and improvement,” in Handbook of partial least squares, V. Esposito Vinzi, W. W. Chin, J. Henseler, and H. Wang, Eds. Springer Berlin Heidelberg, 2010, pp. 47-82.

[31] J. B. Walther, B. Van Der Heide, S. Y. Kim, D. Westerman, and S. T. Tong, “The role of friends’ appearance and behavior on evaluations of individuals on Facebook: Are we known by the company we keep?,” Human Communication Research, vol. 34(1), 2008, pp. 28–49.

[32] I. M. Woon and L. G. Pee, “Behavioral factors affecting Internet abuse in the workplace-an empirical investigation,” In Proceedings of the Third Annual Workshop on HCI Research in MIS, Washington, D.C., 2004, pp. 80-84.

[33] J. Wu and K. Y. Goh, “Evaluating longitudinal success of open source software projects: A social network perspective,” In Proceedings of the 42nd Hawaii International Conference on System Sciences, 2009, pp. 1-10.

[34] R. K. Yin, Case study research: design and methods, 3rd ed. SAGE Publications, 2002.