Proceedings of the 8th Annual World Conference
of the Society for Industrial and Systems Engineering,
Baltimore, MD, USA
October 17-18, 2019

Network Structure and the Effectiveness of Crowd-Based Requirements Processes

M. Robinson, S. Sarkani, and T. Mazzuchi

The George Washington University,
Washington D.C., USA

Corresponding author's Email: mrobinson23@gwu.edu

Author Note: Matthew Robinson is a PhD student in Systems Engineering at The George Washington University and a Data Scientist at Capital One. Dr. Shahram Sarkani and Dr. Thomas Mazzuchi are Professors of Systems Engineering and Engineering Management in the School of Engineering and Applied Science at The George Washington University.

Abstract: Proponents argue that crowd-based requirements processes enable project managers to generate better requirements by eliciting feedback from a broader range of stakeholders. However, crowd-sourcing requirements may reduce the ability of systems engineers to respond to stakeholder needs in a timely manner. The inability to address requirements results in unsatisfied stakeholders. It also burdens systems engineers, who need to contend with an unwieldy backlog. This paper tests the hypothesis that crowd sourcing increases the expected close out time for requirements, and the effect of crowd sourcing changes with the structure of the project's stakeholder network. Regression analysis suggests that increasing the proportion of crowd sourced requirements increases the expected close out time for requirements, except for networks with a low degree of localized clustering. The effect of crowd sourcing requirements does not change with network concentration or dispersal. Based on these findings, systems engineers should consider using crowd-based requirements processes for systems with a low degree of localized clustering and apply traditional stakeholder analysis techniques for systems with a high degree of localized clustering.

Keywords: Stakeholder Analysis, Requirements Engineering, Network Analysis, Crowd-Based Requirements Engineering

1. Introduction

The growing prevalence of systems with diverse and geographically distributed stakeholders has stretched the limits of traditional stakeholder analysis techniques. Such conditions make it difficult for systems engineers to identify an exhaustive list of stakeholders at the onset of a project. This leads to an unclear understanding of stakeholder needs, and negatively impacts the quality of system requirements. In order to deal with these challenges, the systems engineering community has begun to develop techniques—known as crowd-based requirements processes—for eliciting requirements from crowds of stakeholders.

Proponents of crowd-based requirements processes argue that engagement with crowds of stakeholders enables systems engineers to better understand the range of potential use cases for a system. This leads to better defined system requirements and more informed prioritization decisions. Difficulties arise, however, because crowd sourcing can overwhelm systems engineers with a large volume of requirements, leading to requirements that languish in the backlog and go unaddressed for a long period of time. Unaddressed requirements result in disappointed stakeholders who have a higher propensity to disengage from the project.

This paper hypothesizes that crowd sourcing increases the expected close out time for requirements, and that the impact of crowd sourcing changes with network structure. The data set consists of 564 open source software projects from GitHub. Each project has a corresponding stakeholder network for which the Gini coefficient measures network concentration, the average minimum path measures network dispersion, and the clustering coefficient measures localized clustering. Regression analysis suggests that crowd sourcing increases the expected close out time for requirements in systems with a high degree of localized clustering but decreases it for systems with a low degree of localized clustering. The concentration and distribution of the stakeholder networks have no impact on the expected requirement duration. Analysis suggests that systems engineers should consider crowd-based requirements processes for systems with a low degree of localized clustering but may prefer traditional stakeholder analysis techniques for systems with a high degree of localized clustering.

ISBN: 97819384961-7-2

2. Literature Review

Groen et al. (2017) acknowledges that, while crowd-based requirements processes facilitate the collection of feedback from a broad array of stakeholders, the volume of feedback presents challenges for project managers. Specifically, project managers have difficulty identifying groups of stakeholders with common interests. They also tend to overlook the needs of less active stakeholders. In addition, Groen and his colleagues identify the need to conduct empirical studies to evaluate the benefits of crowd-based requirements processes and how they weigh against the potential drawbacks. By evaluating the impact of network structure on expected requirement close-out time, this paper provides systems engineers with an empirical basis on which to decide whether to crowd source requirements for a given project.

Previous research has explored how network structure effects requirements processes for open source software. Specifically, Lyytinen and Iyers (2019) claim that open source projects with more centralized stakeholder networks have higher task completion rates and find evidence to support their hypothesis. Lyytinen and Iyers' research differs from this paper because their metrics capture network concentration but do not consider network dispersion or localized clustering. In addition, they focus on task completion velocity rather than close out time for requirements. Nevertheless, Lyytinen and Iyers' provide a solid foundation for the analysis in this paper and build confidence in the notion that network structure affects the quality of project management processes. This paper continues Lyytinen and Iyers' line of research by linking it to the literature on crowd-based requirements processes and using different measures for project management effectiveness and network structure.

Linaker et al. (2019) develop a methodology that builds stakeholder networks from project management data by creating a node for each stakeholder and connecting any two stakeholders who have collaborated on a requirement. This paper constructs stakeholder networks using the same methodology. Linaker's paper uses centrality measures for the stakeholder network to help firms develop strategies for influencing the prioritization of requirements for open source software. By contrast, this research focuses on project level metrics. In addition, Linaker's work considers open source projects that include considerable influence from large corporations and government agencies, using Apache Hadoop as a case study. This paper instead concerns projects with open project governance structures.

3. Methodology

3.1 Data

The data for this research consists of publicly available project management data from GitHub, a widely used code repository and collaboration tool. Stakeholders in a GitHub project fall into two categories: contributors and users. Contributors include any stakeholder who has committed code to a project. Users include any stakeholder who benefits from the project but does not contribute to the code base. Although users do not write source code for a project, they participate in the software development process by providing feedback and submitting requirements. If a non-contributor submits a requirement, this research considers the requirement crowd sourced.

GitHub projects manage requirements through issues. Collaboration on issues occurs through comments, which also serve as documentation. GitHub tracks both tasks and pull requests as issues. Since pull requests represent contributions to the code base rather than requirements, the analysis in this paper excludes pull requests. GitHub issues present additional difficulty because in some cases, issues function as a help forum rather than a project management artifact. Fortunately, GitHub provides labels to separate questions from project tasks. In order to limit the analysis to project requirements, the analysis only considers issues with the labels "bug", "change", "enhancement", "feature", "feature request", or "suggestion" and ignores issues with the labels "documentation", "help wanted", and "question".

In total, the data set consists of 564 packages from a curated list of popular open source projects, 133,738 distinct users, 324,994 issues, and about 1.28 million comments. The histogram in Figure 1 confirms that the proportion of requirements sourced from the crowd varies considerably in the data set. In addition, the data set includes a wide range of values for expected requirement duration. For most projects, users can expect the project team to close out a requirement within three months, although the distribution has a heavy right tail.

ISBN: 97819384961-7-2

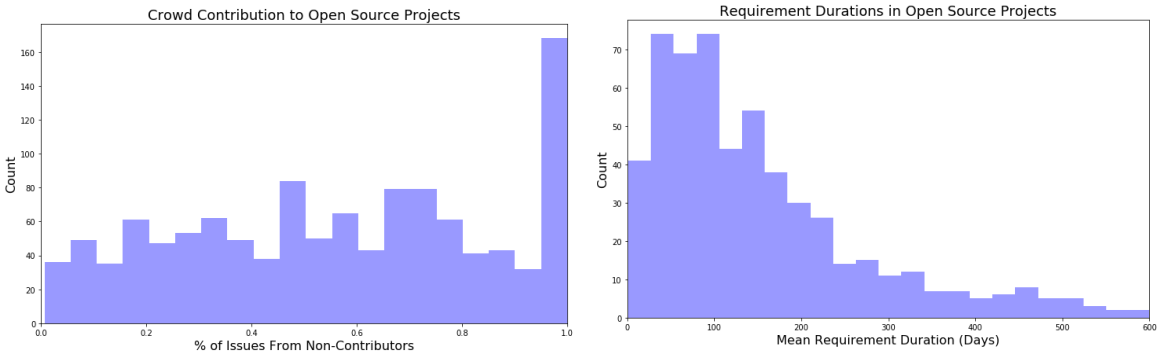

Figure 1. Summary Statistics for the GitHub Data Set

3.2 Stakeholder Networks

Per Linaker's methodology for constructing stakeholder networks, each node represents a stakeholder and an edge connects any two nodes who have collaborated on a requirement. Within the context of the GitHub data set, this means that two stakeholders have commented on the same issue. The structure of the networks within the data set varies considerably. From left to right, Figure 2 depicts three stakeholder networks from the data set that capture the three network structural measures. The first shows a project with strong localized clustering, the second project has a high level of concentration, and the third project has high dispersal. The diversity of the networks in the data set provides an opportunity to test the impact of different configurations on the effectiveness of crowd sourcing.

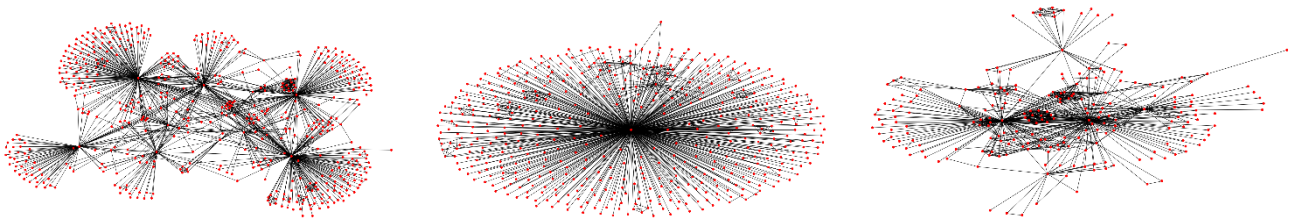

Figure 2. Stakeholder Networks for pyjwt, i18next, and aws-sdk-php

The Gini coefficient measures the amount of inequality in the degree distribution of the network and characterizes the level of concentration in the network. Although researchers typically use the Gini coefficient as a measure of income inequality, precedent exists for using it as a measure of concentration in social networks, including in Toral, Martinez-Torres, and Barrero's 2010 paper. The average minimum path between nodes in the network provides a measure of the network's dispersal. Watts and Strogatz (1998) describe networks with short average minimum paths as small world networks. The clustering coefficient computes the probability that two incident nodes will form a triangle and measures the degree of localized clustering in the network. The regression analysis in this research tests the effect of these structural measures on the effectiveness of crowd-based requirements processes.

3.3 Regression Analysis

Hypothesis testing relies on a linear regression model with mean requirement close out time as the dependent variable. The independent variables include the percentage of requirements a project sources from the crowd, the network structure variables, and several control variables.

To ensure proper model specification, the regression also includes several control variables. Excluding these variables would produce sub-optimal regression results because factors in the error term would correlate with both the independent variable and the covariates. Specifically, the regression incudes the variance of requirement close out times, the total number of contributors to the project, and the age of the project. Duration variance and project age both correlate positively with

duration mean due to the existence of more opportunities for longer close out times. The variable for total contributors correlates negatively with duration mean, reflecting projects with increased work capacity.

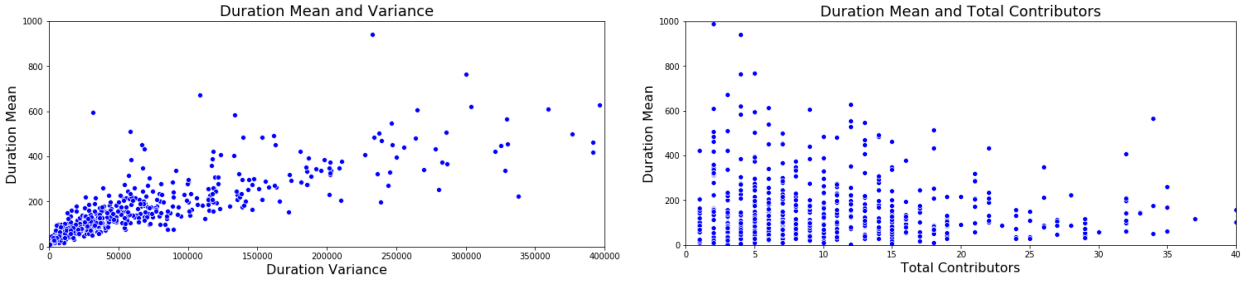

Figure 3. Additional Important Variables

4. Results

4.1 Regression Models

Estimating a simple ordinary least squares (OLS) model using mean requirement duration as the dependent variable and the percentage of crowd sourced requirements as the independent variable shows that crowd sourcing has a positive and statistically significant effect on mean requirement duration, as expected. Using the network variables as the independent variables shows that the clustering coefficient and average minimum path have a statistically significant impact on mean requirement close out time, while the Gini coefficient does not. In this model, increasing localized clustering and network dispersal both increase expected requirement close out time. These initial results build confidence in the notion that both crowd sourcing and network structure have an impact on expected requirement duration.

Table 1. Regressions on Expected Requirement Duration

Variables	OLS	OLS	OLS	OLS	Robust
Adj. R2	0.056	0.079	0.721	0.728	0.728
Intercept	96.25***	-143.37***	-48.41	42.54	42.54
Crowd Pct	1.43***		0.57***	-1.28**	-1.28**
Gini Coef.		-129.98	45.83		
Cluster Coeff		192.09***	100.01***	-30.14	-30.14
Cluster X Crowd				3.14***	3.14***
Avg Min Path		121.74***	-4.05		
Duration Var.			0.0012***	0.0012***	0.0012***
# of Contributors			-0.64*	-0.45	-0.45**
Project Age			6.09***	5.90***	5.90***

Significance Levels: *** 1%, ** 5%, * 10%

The independent variables in the next regression include the percentage of crowd sourced requirements, the Gini coefficient, the clustering coefficient, the average minimum path, and the control variables. The results of this regression appear in the third column of Table 1. Of these variables, all have a statistically significant impact on mean requirement duration except for the Gini coefficient and the average minimum path. The results of this regression suggest that network concentration and dispersal do not affect mean requirement close out time, given the other factors in the regression.

The final regression removes the insignificant network measures and adds an interaction term between crowd percentage and the cluster coefficient. The results of this regression appear in the fourth column of Table 1. In this regression, the total effect of crowd percentage depends on the value of the cluster coefficient and has a magnitude of $\alpha_{total} = \beta_{crowd} + \beta_{cluster} * cluster$. The median value for the cluster coefficient (0.62) implies a total effect of 0.68 for crowd percentage. On

average, crowd sourcing an additional one percent of requirements adds 0.62 days to the expected requirement close out time. In other words, at the median value for the clustering coefficient, crowd sourcing an additional one percent of requirements adds about as much time to the expected requirement duration as removing one contributor from the project, which adds 0.45 days on average.

Although crowd sourcing has a strong positive effect on requirement duration at the median level of localized clustering, the direction of the effect reverses for networks with a clustering coefficient below 0.35. In extreme cases where no localized clustering exists, increasing the percentage of crowd sourced requirements by once percent reduces the expected requirement duration by 1.35 days. This analysis suggests that crowd sourcing reduces expected requirement close out time for projects with a low degree of localized clustering and increases expected requirement close out time for projects with a high degree of localized clustering.

4.2 Regression Diagnostics

The validity of the results presented above require the model to conform to the Gauss-Markov assumptions. Per Wooldridge's (2015) formulation, the Gauss-Markov assumptions include (1) linearity with respect to parameters, (2) a randomly sampled data set, (3) no perfect collinearity among the independent variables, (4) zero conditional mean for the error term, and (5) homoskedasticity. The adjusted R2 (0.728) of the final OLS model suggests a good model fit, making (1) a reasonable assumption. Assumption (2) effectively requires project management decisions for a give project to have no impact on the other projects in the data set. Due to the multitude of contributors for most projects and the limited overlap between projects, the assumption of independence seems reasonable.

The variance inflation factor (VIF) for each independent variable helps determine the amount of multicollinearity in the model. Each variable that does not include an interaction term has a VIF of less than two, which suggests that the model does not suffer from excessive multicollinearity. Crowd percentage and the interaction term between crowd percentage and clustering coefficient do have high VIF values at 17.28 and 22.08, respectively. However, the multicollinearity between these variables results from the construction of the interaction term and does not pose a major concern. The VIF for crowd percentage falls below two in the regression that omits the interaction term. These results suggest that the model does not suffer from a multicollinearity problem, making (3) a reasonable assumption.

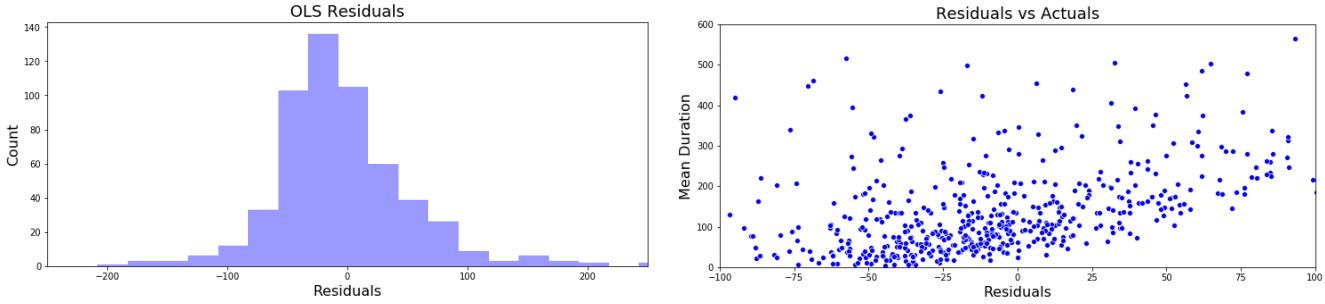

Figure 4. Residual Plots

Assumption (4) implies an error term with zero conditional mean. Plotting a histogram of the residuals shows an approximately normal distribution centered around zero. Computing the mean of the residuals results in a value within floating point tolerance of zero. Based on this analysis, assumption (4) holds. Under assumptions (1) through (4), the OLS model produces unbiased estimates for the parameters. In addition, despite not introducing any restriction on the domain of the dependent variable, the model does not produce any negative predicted values, which bolsters confidence in the fit of the model.

Although assumptions (1) through (4) guarantee that the OLS model produces unbiased estimates, heteroskedasticity remains a concern. Specifically, the plot of residuals against actuals reveals that few observations with actual values of less than 100 have positive residuals. Consequently, the variance of the residuals decreases with the value of the dependent variable. Since the variance of the residuals does not remain constant, this suggests the presence of heteroskedasticity. A Breusch-Pagan test for heteroskedasticity supports this observation and rejects the null hypothesis of homoskedasticity at the one percent significance level. As a result of this finding, inference requires heteroskedasticity robust standard errors, which appear in the fifth column of Table 1. Since assumptions (1) through (4) hold, the parameters from the regression remain valid. Under the robust standard errors, the number of contributors becomes statistically significant at the five percent level, whereas it had no

statistical significance in the original OLS model. Based on a holistic view of the model diagnostics, the OLS model with heteroskedasticity robust standard errors appears valid and supports the conclusion that crowd sourcing increases expected requirement duration for projects with a high degree of localized clustering, and reduces it for projects with a low degree of localized clustering.

5. Conclusions and Future Research

The results of this analysis show that, on average, projects that source more requirements from the crowd take longer to address requirements. Fixing the clustering coefficient to its median value, the analysis shows that increasing the percentage of requirements sourced from the crowd increases the expected close out time for a requirement by the same amount as removing a contributor from the project—about a half day in both cases. The magnitude of the effect changes with the degree of localized clustering in the network. Crowd sourcing has the largest detrimental impact on expected close out time for stakeholder networks with a high degree of localized clustering. The size of the impact decreases as localized clustering decreases, and reverses (i.e. improves requirement close out time) for projects with a clustering coefficient of less than 0.35. Network concentration and dispersal do not have a statistically significant impact on the effect of crowd sourcing.

From a practical point of view, these results suggest that systems engineers should consider using traditional stakeholder analysis techniques for projects that demonstrate a high degree of localized clustering. Using crowd-based requirements processes for such systems result in longer requirement close out times. For these projects, systems engineers have difficulty maintaining a growing backlog. Meanwhile stakeholders grow frustrated at the project's inability to address their needs in a timely manner. In addition, a high degree of localized clustering lends itself well to traditional stakeholder analysis techniques, because it implies the existence of well-formed groups of stakeholders, who likely share the same concerns.

By contrast, crowd-based requirements processes may benefit systems engineers who work on projects that demonstrate less localized clustering. For such systems, generating more requirements from the crowd decreases the expected close out time for requirements. Traditional stakeholder analysis techniques may not perform as well without localized clustering because systems engineers have difficultly classifying and grouping stakeholders. The problem multiplies if the system has a multitude of stakeholders. As such, crowd-based requirements processes provide a good option for systems with a low degree of localized clustering.

The analysis in this research only considers the expected close out time for requirements. Systems engineers may have an interest in other project-level metrics when determining a strategy for requirements generation. Follow-on research will evaluate the other proposed benefits and drawbacks of crowd-based requirements processes. Specifically, future research will address the impact of crowd sourcing and network structure on the diversity of requirements, test whether crowd-members and contributors submit different requirements and evaluate the quality of crowd-sourced requirements. Analyzing these additional factors will enable the development of a framework that allows systems engineers to evaluate the trade-offs involved in the use of crowd-based requirements processes.

6. References

Fahrmeir, L., Kneib, T., Lang, S., & Marx, B. (2013). *Regression: models, methods and applications*. Springer Science & Business Media.

Gerth, R. J., Burnap, A., & Papalambros, P. (2012). *Crowdsourcing: A primer and its implications for systems engineering*. MICHIGAN UNIV ANN ARBOR.

Groen, E. C., Seyff, N., Ali, R., Dalpiaz, F., Doerr, J., Guzman, E., ... & Stade, M. (2017). The crowd in requirements engineering: The landscape and challenges. *IEEE software*, *34*(2), 44-52.

Hosseini, M., Phalp, K. T., Taylor, J., & Ali, R. (2014). Towards crowdsourcing for requirements engineering.

Iyer, D. G., & Lyytinen, K. (2019). Requirements Engineering Effectiveness in Open Source Software: The Role of Social Network Configurations And Requirements Properties. *Requirements Engineering, 5,* 15-2019.

LaToza, T. D., & Van Der Hoek, A. (2015). Crowdsourcing in software engineering: Models, motivations, and challenges. *IEEE software, 33*(1), 74-80.

Levy, M., Hadar, I., & Te'eni, D. (2015, August). A gradual approach to crowd-based requirements engineering: The case of conference online social networks. In *2015 IEEE 1st International Workshop on Crowd-Based Requirements Engineering (CrowdRE)* (pp. 25-30). IEEE.

Lim, S. L., & Ncube, C. (2013, June). Social networks and crowdsourcing for stakeholder analysis in system of systems projects. In *2013 8th International Conference on System of Systems Engineering* (pp. 13-18). IEEE.

Linåker, J., Regnell, B., & Damian, D. (2019). A method for analyzing stakeholders' influence on an open source software ecosystem's requirements engineering process. *Requirements Engineering,* 1-16.

Robinson, W., & Vlas, R. (2015). Requirements evolution and project success: an analysis of SourceForge projects.

Snijders, R., Dalpiaz, F., Hosseini, M., Shahri, A., & Ali, R. (2014, December). Crowd-centric requirements engineering. In *2014 IEEE/ACM 7th International Conference on Utility and Cloud Computing* (pp. 614-615). IEEE.

Toral, S. L., Martínez-Torres, M. D. R., & Barrero, F. (2010). Analysis of virtual communities supporting OSS projects using social network analysis. *Information and Software Technology, 52*(3), 296-303.

Watts, D. J., & Strogatz, S. H. (1998). Collective dynamics of 'small-world' networks. *Nature, 393*(6684), 440.

Wood, J., Sarkani, S., Mazzuchi, T., & Eveleigh, T. (2013). A framework for capturing the hidden stakeholder system. *Systems engineering, 16*(3), 251-266.

Wooldridge, J. M. (2015). Introductory econometrics: A modern approach. Nelson Education.

Yitzhaki, S. (1979). Relative deprivation and the Gini coefficient. *The quarterly journal of economics*, 321-324.