# Examining Turnover in Open Source Software Projects Using Logistic Hierarchical Linear Modeling Approach

Pratyush N Sharma[1], John Hulland[2], and Sherae Daniel[1]
1 University of Pittsburgh, Joseph M Katz Graduate School of Business,
229 Mervis Hall, Pittsburgh, 15232, USA
{pns9,sld54}@pitt.edu

2 University of Georgia, Terry College of Business,104 Brooks Hall, 310
Herty Drive, Athens, 30602, USA
jhulland@uga.edu

**Abstract.** Developer turnover in open source software projects is a critical and insufficiently researched problem. Previous research has focused on understanding the developer motivations to contribute using either the individual developer perspective or the project perspective. In this exploratory study we argue that because the developers are embedded in projects it is imperative to include both perspectives. We analyze turnover in open source software projects by including both individual developer level factors, as well as project specific factors. Using the Logistic Hierarchical Linear Modeling approach allows us to empirically examine the factors influencing developer turnover and also how these factors differ among developers and projects.

**Keywords:** Open Source Software, Turnover, Logistic Hierarchical Linear Modeling.

## 1 Introduction

Developer turnover in open source software (OSS) projects is a nontrivial issue because of the frequency with which it occurs and the difficulties new developers face in contributing to a project. Robles and Gonzales-Barahona [8] analyzed the evolution of some popular OSS projects (such as GIMP, Mozilla etc.) over a period of 7 years and found that these projects suffered from yearly turnover in core development teams and had to rely heavily on regeneration. Turnover is a critical problem in software development projects because it can lead to schedule overruns [1] and regenerating teams is a complicated issue [7]. A majority of the OSS research concerns itself with a developer's motivation to contribute to OSS development [2; 3].

Prior studies have tended to focus on the explanation of developer activity levels using either the individual perspective [2; 3] or the project perspective [10]. However, since OSS participants are embedded in projects it is important to relate characteristics of individuals and the characteristics of projects in which they function. Disaggregating all project level variables in an individual level analysis may lead to the violation of the assumption of independence of observations, since all developers will have the same value on each of the project variables. On the other hand, aggregating developer level variables to a project level analysis may lead to unused within group information [6]. None of the research studies have attempted to model turnover behavior in OSS in a comprehensive fashion taking into account

both the developer level and project level factors. In order to address these limitations and expand the existing research, this exploratory study develops a model of turnover behavior in OSS by focusing on two levels: the developer level, which examines factors that may affect developers' decisions to become inactive, and the project level, which examines the factors that may influence the rates of turnover among projects.

# 2 Methodology

To explore and explain the nature and impact of a developer and project variables on turnover, we used archival data. The sample of projects and participants was drawn from SourceForge (www.SourceForge.net). The sample contained data for 40 currently active projects on SourceForge and 201 developers.

## 2.1 Developer Level Variables

The following five developer level (level 1) variables, including the outcome variable, turnover, were collected –

- Turnover –Turnover was operationalized as a binary outcome variable. A developer was deemed active, coded as 0, if at least one CVS/SVN commit was made by him/her in a 2 month period; otherwise coded as 1. Joyce and Kraut [10] also followed a similar approach in their study of turnover from online newsgroups, however they chose an observation period of six months to determine turnover.
- Role of the Developer –A project may employ developers for various roles that range in the level and kind of expertise required[1]. We created two dummy variables *Developer* and Admin with the base group *Other* (which included all other roles)[2].
- Number of Projects –The number of OSS projects undergoing active development that the developer was involved in.
- Past Activity Level –Past activity was operationalized as a binary variable[3]. A developer was deemed active in the past, coded as 0, if at least one CVS/SVN commit was made by him/her in the previous 10 month period; otherwise, we coded it as 1.
- Tenure – We approximate the tenure of a developer in months by using the date of joining SourceForge.net.

## 2.2 Project Level Variables

The following project level variables (level 2) were collected –

- Project Age – The date the project was registered is available on SourceForge. We calculate the age in number of months since its registration on SourceForge.

---

[1] Some examples of roles developers may perform in the project are as administrators, developers, document writers, project managers, packagers, web designers, etc.

[2] Roughly 25% roles belonged to the *Other* category. Since *Developer* and *Admin* dummies are correlated we also analyzed the data by merging *Other* and *Admin* categories to create a single *Developer* dummy variable. In doing so we found that the HLM results did not change appreciably.

[3] Using a binary dummy variable for measuring turnover and past activity results in loss of variance information and right censoring of data in developer activity levels. Please see the limitations section for how we intend to remedy this problem in the future.

- Size of Project – The number of developers with commit access to the project's CVS/SVN code repository.

### 2.3 Statistical Models and Results

The Hierarchical Linear Modeling (HLM) technique allows researchers to model developer level outcomes within projects and model any between project differences that arise. The study was carried out in two parts and follows the approach recommended by Rumberger [9]. In the first part a developer model of turnover was developed and tested with logistic regression using only developer level variables. This allows an analysis focused only on developer level variables. However, this not only ignores project level variables but also assumes that the effects of developer level variables on turnover do not vary from project to project. This assumption was tested in the second part of the study using logistic HLM analysis. The developer level model used in this part of the study was based on the results of the first part. It allowed us to focus the analysis on explaining between project differences in the predicted mean turnover rates (turnover characteristics adjusted for differences in developer characteristics between projects) and between project differences in the effects of developer level variables on turnover rates.

### 2.4 Logistic Models

A series of linear logistic models were developed and tested to measure the effect of developer level variables on turnover behavior. Turnover is a binary dependent variable that can be expressed as a probability $p_i$, which takes on the value of unity if the developer $i$ becomes inactive in the project, zero otherwise. The probability $p$ is transformed into log of odds (or logit) which is expressed as:

$$Log\,[p_i / (1-p_i)] = \beta_0 + \beta_1 Past\_Activity + \beta_2 Tenure + \beta_3 Developer + \beta_4 Admin + \beta_5 Number\_of\_Projects$$

Table 1 presents the exponentiated logistic coefficients, which represent the ratio of predicted odds of turnover with a one unit increase in the independent variable to the predicted odds without one unit increase. Thus, a value of one signifies no change in the odds of turnover. A value greater than (less than) one indicates that the odds of turnover increase (decrease) due to a unit change in independent variable.

**Table 1.** Predicted odds of turnover

| Variable | Univariate estimates | Multivariate estimates |
|---|---|---|
| Past_Activity | 371.429** | 431.724** |
| Admin | .443* | .989 |
| Developer | 1.104 | .548 |
| Tenure | 1.008 | 1.007 |
| Number_of_Projects | .981 | 1.038 |
| *-2LL* (initial = 266.583) | | 109.008 |
| Cox and Snell $R^2$ | | .543 |
| Nagelkerke $R^2$<br>$\Delta\chi^2 = 157.57$ ($p < .001$) | | .740 |

*p < 0.05, **p < .001

The univariate and multivariate estimates of *Past_Activity* are both significant. The univariate estimate suggests that inactive developers have 371.42% higher odds of turnover than developers that were active. Unsurprisingly, inactive developers did not become active at a later stage. The univariate estimate of *Admin* is also significant and suggests that administrators have 44.3% lower odds of turnover than the *Other* category. This means that administrators are more than twice as likely to remain active than developers with *Other* roles. Since *Past_Activity* and *Admin* were significant in the univariate estimates they were retained for further HLM analysis.

### 2.5 HLM Models

HLM analysis requires two types of models: a level 1 model to estimate the effects of developer level variables on turnover and a level 2 model to estimate the effect of project level variables on the coefficients of the level 1 analysis. We begin the analysis by modeling the unconditional model (base model) with no predictors at either level.

### 2.6 Unconditional Model

$$Log\,[p_{ij}/(1\text{-}p_{ij})] = \beta_{0j}$$

$$\beta_{0j} = \gamma_{00} + u_{0j}$$

This model allows us to ascertain the variability in the outcome variable at each of the two levels i.e. within project and between project variability. The results are shown in Table 2.

**Table 2.** Unconditional Model

| **Fixed effect** | | **Coefficient** | **se** | **p value** |
|---|---|---|---|---|
| Average project mean $\gamma_{00}$ | | .484 | .183 | 0.012 |
| **Random effect** | **Variance component** | **df** | **$\chi$2** | **p value** |
| Project mean, $u_{oj}$ | .314 | 39 | 57.48 | .028 |
| Deviance (-2LL)<br>Estimated parameters | 631.288<br>2 | | | |

The Null hypothesis $H_0$: $\tau_{00} = 0$ is rejected (p = .028). This suggests that significant variation exists among projects in their turnover rates. The intraclass correlation coefficient (ICC) measures the proportion of variance in the outcome that is between projects [11]. ICC values for our analysis suggest that 8.71% variation in turnover that can be explained by level 2 predictors resides between projects. Further, for a project with a typical turnover rate (with $u_{0j} = 0$), the expected log odds of turnover is .484. This corresponds to a probability of 1/ (1 + *e* (.484)) = .38. This means that for a typical developer in a typical project there is a 38% chance of turnover in a 2 month period.

### 2.7 Conditional Model

This model allows part of the variation in the intercept $\beta_0$ (mean turnover rates) to be explained by project level variables (project age and size),

$$Log\,[p_{ij}/(1\text{-}p_{ij})] = \beta_{0j} + \beta_{1j}\,Past\_Activity + \beta_{2j}\,Admin$$

$$\beta_{0j} = \gamma_{00} + \gamma_{01}\,Proj\_Age + \gamma_{02}\,Proj\_Size + u_{0j}$$

$$\beta_{1j} = \gamma_{10}$$

$$\beta_{2j} = \gamma_{20}$$

All the variables were grand mean centered to reduce multicollinearity concerns in group level estimation [5]. Table 3 presents the results of the conditional model.

**Table 3.** Conditional Model

| Fixed effect | Coefficient | se | | p value |
|---|---|---|---|---|
| Average project mean $\gamma_{00}$ | 1.86 | .49 | | 0.001 |
| Proj_Age Slope $\gamma_{01}$ | .005 | .006 | | 0.461 |
| Proj_Size Slope $\gamma_{02}$ | -0.012 | 0.008 | | 0.154 |
| Past_Activity Slope $\gamma_{10}$ | 5.951 | 1.161 | | 0.000 |
| Admin Slope $\gamma_{20}$ | 0.041 | 0.527 | | 0.938 |
| **Random effect** | **Variance component** | **df** | **$\chi$2** | **p value** |
| Project mean, $u_{oj}$ | 0.057 | 37 | 29.48 | >.500 |
| Deviance (-2LL)<br>Estimated parameters | 477.808<br>6 | | | |

The Null hypothesis $H_0$: $\tau_{00} = 0$ fails to be rejected ($p > .500$). This means that after controlling for project size and age no significant variation remains to be explained. The proportion of reduction in variance or variance explained at level 2 is .8184, implying that project size and age account for 81.84% of the explained variance at level 2. The Deviance (-2 Log Likelihood) is also significantly improved from the base model ($\Delta D = 153.48$, $\chi 2_{df} = 4$, $p < .001$), suggesting a good model fit and a fully identified model[4].

## 3 Limitations & Future Directions

Like all empirical work this study is limited in many ways. First, the sample is biased toward more active projects. Such projects may have well developed infrastructures allowing retention of active members and/or a constant inflow of newer active members. Including less active projects in the future should allow for more robust and generalizable results. Second, the use of binary variables for turnover and past activity leads to loss of variance information and right censoring of the data. To address this critical issue in the future, we will rely on techniques such as survival modeling that allows inference from right censored data. Finally, we will seek a conceptual integration of developer and project level factors in modeling turnover rather than just an empirical integration.

## 4 Conclusion

In this preliminary study, we argued that taking both the developer and the project level factors into account will lead to a richer understanding of the issue of turnover in open source projects. Our analysis suggests that past activity, developer role, project size and project age are important predictors of turnover. We find that there exists a significant variation in mean turnover rates among projects on SourceForge and that project age and project size account for a sizable proportion of this variation.

4 A conditional model that included all developer level variables did not further improve deviance and was rejected in favor of the more parsimonious model presented here.

## 5 References


1. Collofello, J., Rus, I., Chauhan, A., Smith-Daniels, D., Houston, D., and Sycamore, D.M. 1998. "A System Dynamics Software Process Simulator for Staffing Policies Decision Support," in *Proceedings of the Thirty-First Hawaii International Conference on System Sciences*, Hawaii.
2. Hars, A., and Ou, S. 2002. "Working for Free? Motivations for Participating in Open Source Projects," *International Journal of Electronic Commerce* (6:3), pp. 25-39.
3. Hertel, G., Niedner, S., and Herrmann, S. 2003. "Motivation of Software Developers in Open Source Projects: an Internet-Based Survey of Contributors to the Linux Kernel," *Research Policy* (32), pp. 1159-1177.
4. Joyce, E., and Kraut, R.E. 2006. "Predicting Continued Participation in Newsgroups," *Journal of Computer Mediated Communication* (11), pp. 723-747.
5. Raudenbush, S.W. 1989. "Centering Predictors in Multilevel Analysis: Choices and consequences," *Multilevel Modeling Newsletter* (1), pp. 10-12.
6. Raudenbush, S.W. and Bryk, A.S. 2002. *Hierarchical Linear Models* (Second Edition), Thousand Oaks: Sage Publications.
7. Reel, J.S. 1999. "Critical Success Factors In Software Projects," *IEEE Software* (16:3), pp. 18-23.
8. Robles, G., and Gonzalez-Barahona, J.M. 2006. "Contributor Turnover in Libre Software Projects," in *IFIP International Federation for Information Processing: Open Source Systems*, Springer, Boston.
9. Rumberger, R.W. 1995. "Dropping Out of Middle School: A Multi-Level Analysis of Students and Schools," *American Educational Research Journal* (32:3), pp.583-625.
10. Stewart K.J., Ammeter, T.A. and Maruping, L. 2006. "Impacts of License Choice and Organizational Sponsorship on User Interest and Development Activity in Open Source Software Projects," *Information Systems Research* (17:2), pp. 126-144.
11. Snijders, T. and Bosker, R. 1999. *Multi Level Analysis: An Introduction to Basic and Advanced Multilevel Modeling*, Sage Publications.