What Community Contribution Pattern says about Stability of Software Project?

Ayushi Rastogi  
IIIT-Delhi  
Email: ayushir@iiitd.ac.in

Ashish Sureka  
IIIT-Delhi  
Email: ashish@iiitd.ac.in

Abstract—Free / Libre Open Source Software community management is a big issue. Contributor churn (join or leave the project) causes failure of the majority of software projects and is inevitable. In this study, we present a framework that characterizes stability of the community in software maintenance projects on the community participation patterns by mining Issue Tracking System. We identify key stability indicators and propose metrics to measure them. We conduct time series analysis on metrics data to examine the stability of the community. We model community participation patterns and forecast future behavior to help plan and support informed decision making. We present a case study of four years data of Google Chromium Project and investigate the inferential ability of the framework.

I. RESEARCH MOTIVATION AND AIM

Research shows that majority of Free / Libre Open Source Software (FLOSS) projects fail due to lack of sustained contributors [1][2][3][4][5][6]. Lack of sustained contributors in FLOSS project cause schedule overrun [7] and team regeneration [6] which adversely affects the existence and quality of the project [4][8].

In FLOSS project contributors join and leave the project (contributor churn) at will [6]. The voluntary participation in FLOSS project makes it hard to understand contributor churn [2] thereby escalating the challenge of estimating its effects. The inevitable and non-trivial nature of estimating the effect of contributor churn makes it hard to characterize the stability of software maintenance projects. So the research aim of the work presented in this paper is:

1) To investigate metrics to objectively characterize community stability on key stability indicators by mining Issue Tracking System.

2) To demonstrate the inferential ability of time series data on key stability indicators to investigate the stability of the community and estimate future contribution to support informed decision making.

II. RELATED WORK AND RESEARCH CONTRIBUTIONS

In software lifecycle FLOSS project undergo phases of contributor evolution. In initial phase small core team develop software. As the project matures over time multiple core teams contribute to develop software [9][10]. Research shows that the majority of FLOSS projects fail due to lack of sustained developers [4][5]. To account for the failure of FLOSS projects, research identifies three community stability indicators, namely attrition, retention, and regeneration [11][4][5][6][12]. However, the studies do not use metrics to quantify contributor churn. In another line of study, research identifies contributors likely to remain by mining software repositories[16][4][15][14]. However, they do not estimate how change in community participation pattern may affect stability of community and organization. Also, we see that Wu et al. and Yu et al. conduct time series analysis of contributor participation to identify trends and use it for prediction [17][18] (brief description in Table I). So the novel contributions of this study are:

1) A framework to quantify key stability indicators on community participation patterns; investigate trends and predict stability of software maintenance projects as identified by mining Issue Tracking System.

2) We propose 3 metrics (based on Key Stability Indicators (KSIs)) for contributors and investigate participation patterns for four roles of contributors, namely reporter, owner, commenter and contributor-cc (cc’ed).

III. RESEARCH METHODOLOGY AND FRAMEWORK

The motivation of the study is to investigate the stability of community in software maintenance projects as identified by Issue Tracking System. We probe temporal community contribution patterns to analyze trends and estimate future participation to support planning and decision making. We study literature to understand Key Stability Indicators in community management. We identify three KSIs namely attrition, regeneration, and retention. In Section IV we propose metrics to quantify the three KSIs by mining ITS. A contributor can play multiple roles by participating in multiple activities. For instance, a contributor may report an issue (reporter), and also own it (owner). A contributor may comment on an issue (commenter) or may collaborate because contributor is cc’ed the issue (collaborator-cc). In this study, we investigate participation in four roles, and present the results. The list of four roles is representative and not exhaustive. Also the roles are overlapping and by no means represent disjoint sets. We generate time series data of metrics and investigate the inferential ability of the temporal trends. However, the metrics

<table>
<thead>
<tr>
<th>Start Date</th>
<th>End Date</th>
<th>Available Reports</th>
<th>Bug Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>2009-01-01</td>
<td>2009-12-31</td>
<td>025841</td>
<td>13006</td>
</tr>
<tr>
<td>2010-01-01</td>
<td>2010-12-31</td>
<td>044041</td>
<td>26634</td>
</tr>
<tr>
<td>2011-01-01</td>
<td>2011-12-31</td>
<td>051683</td>
<td>28170</td>
</tr>
<tr>
<td>2012-01-01</td>
<td>2012-12-31</td>
<td>057970</td>
<td>30510</td>
</tr>
</tbody>
</table>

TABLE II: Experimental Dataset
IV. STABILITY CHARACTERIZATION METRICS

Attrition, Regeneration, and Retention are three KSIs. In this section, we propose metrics, and compute results. Each measure of a metric uses data from two consecutive time intervals as input. Thus we generate time series data for 15 time intervals (out of 16 data points for each metric measured quarterly) for analysis. We explore the inferential ability of time series data through research questions that may help project managers and decision makers in informed decision making.

A. Attrition Rate

In maintenance phase of software development lifecycle projects loose its critical human resource. If the contributor does not participate in one time-interval we assume that contributor has left the project. The contributor may be regenerated (substituted) or lost [10]. Attrition Rate (AR) for time interval $T_t$ measures in percentage the fraction of contributors who left the project in time interval $T_t$ divided by cardinality of all participants in FLOSS projects who were in project in time $[t-1]$.

$$|AR_{T_t}| = \frac{|Con_{T_{t-1}} \setminus Con_{T_t}|}{|Con_{T_{t-1}}|} \times 100 \quad \text{where } t > 1 \quad (1)$$

B. Regeneration Rate

FLOSS projects evolve temporally with contributors entering or leaving the project at will. Research shows that new entrants drive a large fraction of open source projects as original contributors leave the project [9]. On one hand, Regeneration Rate accounts for replenishing resources lost due to contributor attrition, and to meet additional resource requirement in the project. While on the other hand it indicates team composition for any time interval. High regeneration rate indicates imbalanced team composition (junior to senior ratio) with smaller fractions of experienced contributors. The imbalanced team composition may adversely influence the stability of the project. Regeneration Rate (RgR) for time interval $T_t$ measures the rate (in percentage) at which new contributors start participating in ITS in two consecutive time intervals under analysis. In set notations, we define regeneration rate for time interval $T_t$ as fraction of cardinality of set difference of contributors (Con) in $T_t$ from contributors (Con) in time interval $T_{t-1}$ to the cardinality of contributors (Con) in time interval $T_t$ multiplied by 100. The unit of measurement is quarter$^{-1}$.

$$|RgR_t| = \left| \frac{Con_{T_{t-1}} \setminus Con_{T_t}}{Con_{T_t}} \right| \times 100 \quad \text{where } t > 1 \tag{2}$$

C. Retention Rate

Knowledge acquired with experience based on formal and informal understanding of the project cannot be transferred [4]. When a contributor leaves project knowledge acquired goes with the contributor, causing knowledge loss. A key parameter to understand the stability of the software maintenance project is to know its ability to retain knowledge or its contributors. Retention Rate measures contributors retained by the project. Retention Rate (RtR) for time interval $T_t$ measures percentage of contributors retained out of all participants in time interval $T_{t-1}$. In set notations, Retention Rate for time interval $T_t$ is the ratio of cardinality of intersection of contributors (Con) in time interval $T_t$ and preceding time interval $T_{t-1}$ to the cardinality of contributors (Con) in time interval $T_{t-1}$. The unit of Retention Rate is quarter$^{-1}$.

$$|RtR_t| = \left| \frac{Con_{T_t} \cap Con_{T-1}}{Con_{T_{t-1}}} \right| \times 100 \quad \text{where } t > 1 \tag{3}$$

The three metrics proposed in the study cannot be compared. So to help cross comparison, we normalize the metrics on the union of contributor count in two consecutive time intervals $T_t$ and $T_{t-1}$. The normalization facilitates comparison at the cost of affecting the scores. In this study we conduct analysis of modified metrics to ease comparison. The three metrics can be restated as:

$$|AR_t| = \left| \frac{Con_{T_{t-1}} \setminus Con_{T_t}}{Con_{T_{t-1}} \cup Con_{T_t}} \right| \times 100 \quad \text{where } t > 1 \tag{4}$$

Together the three metrics sum to 100 as shown below:

$$AR + RgR + RtR = 100 \tag{7}$$

Figure 2, Figure 3, and Figure 4 show Attrition Rate, Regeneration Rate, and Retention Rate respectively for 15 time intervals starting from ‘2009-04:06’. The horizontal axis of the plot represents consecutive time intervals measured quarterly and the vertical axis shows metric score in percentage. Colored lines (refer to the legend) present metric scores for contributors and four roles of the contributor (reporter, owner, cc’ed and commenter).

In the next section, we investigate temporal community participation patterns (as measured by the three metrics) to comprehend the stability of the project. We present insights as research questions (RQs) that establish correlation, though not the causation, between temporal community participation patterns, and stability of the community.
D. Observations

**RQ1: Do we observe high Attrition Rate in Google Chromium Project?**
In Figure 2, contributor Attrition Rate in Google Chromium Project (shown in black) ranges from 27% to 47% with 34.7% mean and 4.07 % standard deviations. It implies that every three months one-third of the contributors discontinue to participate in ITS. The high attrition rate in GC-ITS raises concerns regarding the stability of the project. However, it fails to give a complete picture of the status of the project. High Attrition Rate makes it relevant to conduct further analysis to understand its cause and hence RQ2.

**RQ2: Do we observe comparable Attrition Rates for all roles? If no, how does it vary with role relevance?**
In Figure 2, we observe a marked difference in Attrition Rates for four roles where role relevance follows the order: owner ≥ cc’ed ≥ commenter ≥ reporter. We observe that Attrition Rate increases with decreasing relevance of the role. We see minimum Attrition Rate for owner (shown in blue) and maximum for reporter (shown in red). This observation follows the intuition and research results of previous research and validates the insight. Attrition Rate for reporter varies from 28.9% to 44.2% with mean 37.7% and standard deviation 4.3%.

The issue count reported in GC-ITS increases over time (refer Figure 1). Thus a high Attrition Rate for the role of reporter presents an open question that is, whether the project is able to compensate for the resources lost and generate resources to meet the increase in requirements. Subsequent RQs addresses such similar concerns.

**RQ3: Is high Attrition Rate accompanied by high Regeneration Rate?**
In Figure 3, we see that in GC-ITS contributor Regeneration Rate fluctuates from 30.0% to 45.4% with mean 38.7% and 4.2% standard deviation. This indicates that the resource lost is regenerated. However, like RQ2, do we observe a marked difference in regeneration rates for four roles and how it may impact the stability. Our next RQ attends anomalous behavior, if any, in Attrition Rate and Regeneration Rate patterns for four roles.
Attrition Rate, Regeneration Rate, and Retention Rate? What
contributors who report bugs.

tion Rate is high for owner and cc'ed compared to retention
23% to 32% with wide variations across four activities. Reten-

time? 

RQ4: Do we observe anomaly, if any, in the temporal
behavior of Attrition Rate and Regeneration Rate for four
roles of contributors?

In Figure 5, we present contribution patterns for four roles of
the contributor. Dotted lines represent actual Attrition Rate
and Regeneration Rate while dark lines indicate smoothed
data (moving average with past 3 observations) to generate
approximate function to capture important patterns. Horizontal
axis show time intervals while vertical axis show scale in
percentage to measure Attrition Rate and Regeneration Rate.
Colors in legend specify the value being measured by dotted
or dark line. In Figure 5 we observe that for the owner
Regeneration Rate is always higher than Attrition Rate
while the same does not hold true for the other three roles.
For reporter, commenter, and cc'ed we observe cross overs
and fluctuations. Cross overs and fluctuations may point to
concerns. Thus, in next RQ we analyze what temporal trend
signals change in the stability of GC-ITS.

RQ5: Do we observe changes in lag between Attrition
Rate and Regeneration Rate?

In Figure 5, we see that for owners, Attrition Rate remains
low and consistent for most parts while a constant dip
in Regeneration Rate is observed. Similarly for other
roles we observe decrease in lag between Attrition Rate
and Regeneration Rate accompanied by crossovers and
fluctuations. Thus, in last few time intervals contributors
lost due to high Attrition Rate is not replenished with high
Regeneration Rate. Also, though we do not observe this trend
for owner their participation pattern is also heading on similar
lines.

RQ6: Do we observe increasing or decreasing trend in
Retention Rate over time?

Figure 4 shows that contributor Retention Rate ranges from
23% to 32% with wide variations across four activities. Reten-
tion Rate is high for owner and cc'ed compared to retention
rate for reporter and commenter. One observation from this
graph is that commenters stay in projects longer than the
contributors who report bugs.

RQ7: Do we observe changes in the relationship between
Attrition Rate, Regeneration Rate, and Retention Rate? What
implications can we draw from such observations?

To answer this RQ, we approximate the trend by fitting a
linear model to the temporal data of Attrition Rate, Regen-
eration Rate, and Retention Rate for 15 time intervals. The
simple linear regression model creates an approximation of the
trend with correlation coefficients $r$ of 0.7314258, -0.7198569,
and -0.7001004 respectively. The fitted model is statistically
significant with p-values 0.004866, 0.008281 and 0.004246
respectively. The fitted model provides an approximation of
the trend. Moreover, the residuals of the model have mean
close of zero and a finite variance indicating that the residuals
do not follow any pattern.

In Figure 6, we observe that Regeneration Rate is
increasing, Attrition Rate is increasing, and Retention Rate
is decreasing. The interpretations that we draw from this
plot are that over time contributors leaving the project is not
regenerated and the retention of contributors is decreasing
over time. In a nutshell, the contributors are slowly moving
away from the project. This observation may influence the
stability of the project depending on whether the project is
in initial phase or has attained maturity. Decreasing contributor
participation during initial phase may indicate decrease in
interest and hence a decrease in popularity of the project.
While if we observe the same trend once the project is mature,
it may indicate that the project observes few issues that must
be incorporated. This observation leads us to the last RQ that
examines the impact of issue count on participation.

RQ8: Do we observe an increase in contribution overhead on
each contributor with time?

In Table II we observe an increase in the contributor count
with increase in issue count. However, the increase is not
proportional. One inference that we draw from the trend is
that over time there is an increase in workload on contributors.
The increase in workload may be welcoming or may further
lead to decrease in contributor participation.

V. ESTIMATING FUTURE PARTICIPATION

Planning is the response of forecasting. Estimating future
participation helps project managers plan ahead and facilitate
informed decision making. Research in FLOSS projects shows
that contributor participation is a volunteering activity where
contributors join or leave project at will. Individual participa-
tion or contribution at microscopic level cannot be estimated.
However, participation at the macroscopic level follows trend
[19].

In this section, we use historical data to make short-term
predictions of contributor participation patterns. The aim is
to explore the time series data of KSI metrics for trends and
investigate the confidence in predicting future participation.
We investigate the correlation between the community partici-
pation pattern and stability of the community. Since we model
historical participation data, the study does not capture changes
in participation pattern due to changes in external factors like
similar project gaining popularity etc. Though these changes
will get reflected in subsequent time intervals.

We propose statistical forecasting models that along with
judgmental forecasting of decision makers (to capture en-
vironmental factors) will present a complete understanding
of factors that influences the stability of the project. We conduct experiments on GC-ITS dataset for 15 time intervals. We enumerate three statistical prediction models, justify the choice, implement, compute accuracy, compare the models for goodness of fit and forecast accuracy and present visualizations and analysis. We present results for the three metrics and visualize one metric. Given the small data size we choose 12 (out of 15) data points for training and the rest 3 for testing. The small dataset influences the selection of modeling technique as discussed below. However, it has limited influence on the accuracy of the prediction as old time series data is less useful to study changes in the project. All experiments present in this section are implemented using toolkits available in R language\textsuperscript{2}.

The baseline model (Model-I) for prediction assumes (also conducted experimentally) that the contributor participation pattern follows normal distribution. We compute mean and standard deviation of the time series data with 99% and 95% confidence interval to estimate participation patterns in following three time intervals (refer Table III). In Table III we see that the simple statistical model computes future participation with 99% confidence interval. Figure 8 show the simple statistical model and the confidence interval for Attrition Rate, Regeneration Rate, and Retention Rate metrics respectively.

The baseline model sets benchmark for the two more involved statistical models. To select models, we conduct exploratory analysis to understand the composition of time series data. In Figure 7, we decompose the time-series data into trends, seasonal components, and remainders using \textit{stl} function in R. \textit{stl} function in R uses loess smoothing to identify seasonal, trend, and irregular components. We use the default settings with seasonality set to periodic. A preliminary analysis of time series data of Attrition Rate metric (also implemented for Regeneration Rate and Retention Rate metrics) suggests that the seasonality (if any) does not grow with the trend. Thus we use an additive model to represent time series data. Figure 7 shows additive components of the time series data along with actual observed data. In Figure 7, we see that Attrition Rate observes an increasing trend with the increase in the irregular component or remainder. The increase in the irregular component with time indicates that the time series does not observe seasonality though we may observe cycles with the trend.

In Figure 7 we observe that the time series data of Attrition Rate metric increases linearly in time. Thus, the model close to the observed trend is a linear regression model. We model the time series data of KSI metrics using \textit{tslm} function. We estimate the goodness of fit of the model and predict accuracy and confidence interval of the observations. The code snippet below from R shows the goodness of fit of the model.

\textbf{Coefficients:}

\begin{tabular}{lcccr}
  \hline
  Estimate & Std. Error & t value & Pr(>|t|) \\
  \hline
  (Intercept) & 29.1656 & 1.6195 & 18.009 & 5.97e-09 \\
  x & 0.7464 & 0.2201 & 3.392 & 0.00686 \\
  \hline
\end{tabular}

Residual standard error: 2.631 on 10 degrees of freedom
Multiple R-squared: 0.535,
Adjusted R-squared: 0.4885

2\textsuperscript{http://www.r-project.org}
From statistical significance testing, p-value less than 0.01 indicates that the linear fit provides a good estimation of the trend. We then use the fitted model to measure the predictive ability of the model with 80% and 95% confidence intervals. Figure 9 show the observed trend, fitted model, and predicted observations along with the confidence intervals. We see in Figure 9 that the fitted model predicts Attrition Rate in next three time intervals with 95% confidence. We observe that the participation varies substantially from the fitted model. In the next model we try to model a better fit to the data (without overfitting) to address this concern.

For statistical time series modeling exponential smoothing and ARIMA models are two popular techniques. The small dataset available for prediction makes estimating parameters for ARIMA modeling difficult without overfitting. Thus ruling out the choice of ARIMA modeling for the study. Exponential smoothing, on the other hand, uses recent observations for prediction. Therefore, for third model we choose exponential smoothing to predict contributor participation patterns. To implement Model III, we use HoltWinters function in R to compute additive exponential smoothing with trend and level and without seasonal components. We forecast participation in three upcoming time intervals with 80% and 95% confidence. Figure 10 show the observed and expected contribution patterns.

A manual inspection of Model-II and Model-III shows slight improvement in prediction accuracy of first time interval (refer Figure 9 and Figure 10) of Attrition Rate metric. However, not much difference is observed for second, and third time intervals. We observe similar trends for Regeneration Rate and Retention Rate metrics. In the code snippet that follows, we compare the prediction accuracies of Model-II and Model-III for different horizons (1, 2 and 3). We conduct Diebold Mariano Test to compare the predictive accuracies of models (implemented in R language). We hypothesize that the predictive accuracy of linear regression model is less than the predictive accuracy of exponential smoothing model. However, the code snippet extracted from R language shows that with forecast horizons 2 and 3 (for Attrition Rate time series data) the two models are comparable in predictive accuracy. However, for forecast horizon 1 exponential smoothing model outperforms linear regression model. We compute the same statistics for Regeneration Rate and Retention Rate metrics and observe same results for the two models.

Diebold-Mariano Test
DM = -1.7906,
Forecast horizon = 3,
Loss function power = 2,
p-value = 0.05044
alternative hypothesis: less

DM = -1.9919,
Forecast horizon = 2,
Loss function power = 2,
p-value = 0.0359
alternative hypothesis: less

DM = -2.8443,
Forecast horizon = 1,
Loss function power = 2,
p-value = 0.00798
alternative hypothesis: less

VI. DISCUSSIONS AND FUTURE WORK

In this section, we state and justify the choices we made in the study, its influence and also discuss the future scope of this work. In this paper, we present a case study on one real-world, large, popular and open source project namely Google Chromium Project. Google Chromium reportedly observed an increase in popularity\(^1\) (with respect to other similar projects) from 2009 to 2012 that is the duration of the study. This increase in popularity is also evident from the increase in issue count reported for the project. The insights present in the study bring out the trends in popular project, however due to limited space availability, we do not bring out the contrast by comparison with projects that heads towards instability or looses its contributors in time. Google Chromium project is mature and

\(^1\)http://statcounter.com
stable. Thus, the trends observed on the Google Chromium project sets benchmark for comparisons and planning.

In this study, we characterize stability of software maintenance projects as identified by ITS. We measure the contribution as activities reported in ITS. Thus, we have no way to measure activities that are not recorded in ITS, or activities that are performed between two reported activities in ITS. Also since we do not consider multiple software repositories that maintains a project we may miss some participation activities for analysis. So the framework present in this study may not measure participation completely. However, assuming that the missing records are uniformly distributed throughout the project, this study provides a fair estimation of the trends which otherwise go unnoticed. The results conducted on one repository namely ITS are encouraging and in future we plan to extend the idea by integrating multiple software repositories. The integrated repositories will present a complete statistical picture to forecast the impact of community participation patterns on stability of the project.

In FLOSS project contributor participation follows Pareto Distribution that is 20% of contributors do 80% of work. Moreover, not all roles are equally relevant. Thus one may argue that it is the contribution pattern of core participants and not all contributors that matters for the project. However, success of FLOSS projects is driven by the masses and not individuals where each contributor has a unique role to serve. For instance, approximately 70% of contributors in FLOSS projects are one time contributors. However, their presence ensures popularity and interest in the project, and is appreciated. So if contributors cease to file issues it is an indicator of decreasing popularity and influences the life time of the project. With decreasing contribution the project diminishes and slowly dies out.

In this study we assume absence of activities in three months as an indicator of contributors leaving the project. This assumption is localized. For instance, an owner who stop participating for three months indicates that the contributor has left the project while a reporter may continue participation even after a year. However, for planning this assumption indicates the trend of participation.

The data available in ITS may not answer all RQs that are relevant to decision makers. However, it gives a justifiable understanding of the stability of the project in a data-driven and objective manner.

Another extension of this work will be to study environmental factors that influence stability of the project by examining contributor participation patterns. We present study on trends of participation, however the inclusion of factors like increasing popularity or market share of similar projects seems to bring promising insights on stability of project.

VII. Conclusions

We present a generalized framework to characterize the stability of software maintenance projects on community participation patterns by mining Issue Tracking System. We propose metrics for three key stability indicators, investigate temporal trends and present our analysis. We model participation pattern, and predict future participation. We propose three statistical prediction models, compare the models for prediction accuracy, and state the inferences. We demonstrate results on four years data of Google Chromium Issue Tracking System and establish the usefulness of the framework. We justify the choices in the study and state future research directions.

REFERENCES

[15] ——, “Does the initial environment impact the future of developers?”