Does Contributor Characteristics Influence Future Participation? A Case Study on Google Chromium Issue Tracking System

Ayushi Rastogi, Ashish Sureka

Does Contributor Characteristics Influence Future Participation? A Case Study on Google Chromium Issue Tracking System

Ayushi Rastogi and Ashish Sureka

Indraprastha Institute of Information Technology-Delhi
{ayushir,ashish}@iiitd.ac.in

Abstract. Understanding and measuring factors influencing future participation is relevant to organizations. This information is useful for planning and strategic decision-making. In this work, we measure contributor characteristics and compute attrition to investigate their relationship by mining Issue Tracking System. We conduct experiments on four year data extracted from Google Chromium Issue Tracking System. Experimental results show that the likelihood of future participation increases with increase in relevance of role in project and level of participation in previous time-interval.

1 Introduction

Contributors leaving project incurs significant direct and indirect costs [1] [2][4][5] thereby making it critical to retain existing contributors [6]. A data-driven approach to study future participation helps overcome challenges in existing practices by providing objectivity and transparency. In this work, we examine contributor characteristics namely role of participation and amount of work done as a measure of predicting future participation. We present an approach that uses statistical measures to classify contributors based on contribution into three mutually exclusive sets namely non-core team, loose core team and tight core team. Also we define attrition as a function of participation in two consecutive time intervals and study attrition rate for four roles (reporter, owner, commenter and cc'ed-contributor (cc'ed)) and three classes of contributors. We conduct experiments on Google Chromium Issue Tracking System (GC-ITS) dataset[1]. The dataset is extracted for four consecutive years and observations are recorded quarterly (3 months). In Issue Tracking Systems contributors play various roles. However, for this work we focus on four roles namely reporter, owner, commenter and cc'ed. These roles are associated with various stages of bug fixing lifecycle. Reporter reports the issue. Issue is fixed by owner in collaboration with commenters participating via threaded discussion forum. Owner may also request participation by cc'ing contributors. Contributors cc'ed are called for to serve specific request in issue.

[1] https://code.google.com/p/chromium

Role	Mean	Std	Min	Max	.25Q	.5Q	.75Q	.9Q	.95Q	.99Q	Skew
Own	24.8	038.1	1	000559	3	10	32	67	94	172.1	003.8
Rep	02.9	008.8	1	000476	1	01	02	04	11	040.0	013.5
CC	20.0	051.4	1	001315	1	04	20	59	92	190.2	009.7
Com	14.0	422.9	1	119803	1	01	02	07	30	239.0	129.9

Table 1: Role based individual contribution pattern

2 Empirical Analysis

Class of contribution Research shows that open source projects follow Pareto Distribution [3] that is 20% of contributors do 80% of work. However, our experimental results do not communicate the same. Table 1 shows that in GC-ITS contribution pattern is highly skewed for four roles. This observation demands statistical and data-driven approach to classify contribution of contributors. In Algorithm 1 we classify each contributor in one of the three mutually exclusive classes namely non-core team, loose core team and tight core team based on contribution. Non-Core Team (NCT) includes contributors who join project to address some specific issue they encountered. Loose Core Team (LCT) includes dedicated contributors with substantial contribution and Tight Core Team (TCT) includes contributors with relatively large contribution (with respect to LCT).

The input to the Algorithm 1 is contribution of contributors where each contributor plays at least one of the four roles. The output is contribution class of contributors calculated for all time-intervals. We measure the contribution for the role of owner, reporter and cc'ed as the total number of issues participated in time-interval defined quarterly. Similarly, for commenter we measure contribution in terms of total number of comments in time-interval. Further to ensure homogeneity for cross comparison we range normalize contribution in each role for all time-intervals on a scale of 1-100 (refer Equation 1). We then append the scores for four roles of contributor for a time-interval to create a structure. All missing values are assigned a negligibly small value (0.0001). We generate cumulative score that measures contribution in terms of relevance of role (weight of owner (W_O) > weight of reporter (W_R) > weight of cc'ed (W_{CC}) > weight of commenter (W_{Cm})) as W_O=0.5, W_R=0.25, W_{CC}=0.125 and W_{Cm}=0.125. The choice of weight depends on specific requirements and may vary for individuals. Assuming that participation in one role is independent of participation in other roles, we use weighted Geometric Mean to generate cumulative score (refer to Equation 2).The score generated ranges from 100 (highest) to approximately 0.0001 (negligible or no contribution). Next we find relative relevance of contribution that is the number of standard deviations datum is related to mean. We calculate Z-Score (refer to Equation 3). If value of Z is less than 0, it indicates that contributor is part of NCT. If value of Z is greater than 1 it defines TCT. Likewise value of Z greater than equal to 0 and less than equal to 1 implies LCT. We calculate contribution class for all time-intervals and return set of contributors for each class for each time-interval.

Algorithm 1 Algorithm to Identify Contribution Class

Require: struct{Owner O, Reporter R, CC'ed CC, Commenter Com} Contributor Con[]
Ensure: ConClass[][3]

procedure ContributionClass(Con)
  for all Time-Interval T_t **do**
    Normalize contribution using Range Normalization [1-100]

$$Y_i = \frac{(100-1) \times X_i}{Max(X) - Min(X)} \tag{1}$$

    Calculate weighted Geometric Mean where $w_O > w_R > w_{CC} >= w_{Com}$ and sum($w_O + w_R + w_{CC} + w_{Com}$) =1

$$Score(S) = O^{w_O} \times R^{w_R} \times CC^{w_{CC}} \times Com^{w_{Com}} \tag{2}$$

    Calculate Z-Score

$$Z = \frac{S - \mu}{\sigma} \tag{3}$$

    if $Z < 0$ **then**
      $ConClass_{T_t}[1] \leftarrow$ Con[Z< 0] ▷ Non-Core Team
    else if $Z > 1$ **then**
      $ConClass_{T_t}[2] \leftarrow$ Con[Z > 1] ▷ Tight Core Team
    else
      $ConClass_{T_t}[3] \leftarrow$ Con[Z >= 0 && Z <= 1] ▷ Loose Core Team
    end if
    ConClass[T_t] $\leftarrow ConClass_{T_t}$
  end for
  return ConClass
end procedure

Attrition Rate In this study, we believe that the contributor has left the project if duration of inactivity exceeds one time-interval (in this case measured quarterly). Thus Attrition Rate (AR) for time-interval T_t measures (in percentage) the fraction of contributors who left the project in time-interval T_t to the total number of contributors who participated in time-interval T_t and its preceding time-interval T_{t-1} .

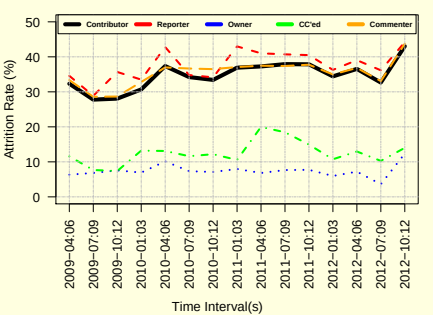

Fig. 1: Attrition rate of maintainers for four years

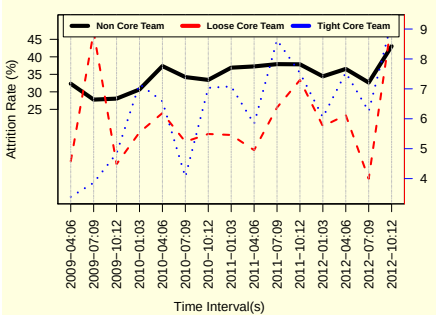

Fig. 2: Comparison of attrition rates for contribution classes for four years

Graphical Analysis of the relationship between Contributor Characteristics and Future Participation In Figure 1 horizontal axis of the plot represents consecutive time-intervals (measured quarterly) and vertical axis shows attrition rate. Colored lines (refer to legend) present attrition rate for contributors and their roles. We observe that contributor attrition rate (irrespective of roles as shown in black) fluctuates from 27% to 47%. Also we observe marked difference in attrition patterns for four roles. We see minimum attrition rate for owner (shown in blue) and maximum for reporter (shown in red). This follows the intuition that not every contributor can own issues. Figure 2 compares attrition rate of three classes of contributors namely non-core team, loose core team and tight core team (refer Algorithm 1) across four years. We see in Figure 2 that the attrition rate for LCT and TCT ranges from 3% to 10% which is relatively less than the attrition rate for NCT (ranges between 27% and 43%). It indicates that retention in project is directly related to degree of involvement in project. Also interestingly after initial fluctuations, attrition rate of TCT is higher than attrition rate of LCT indicating that TCT contributes relatively large however sporadically.

3 Acknowledgement

The work presented in this paper is supported by TCS Research Fellowship for PhD students awarded to the first author. The author would like to acknowledge Dr. Pamela Bhattacharya for useful insights and inputs.

References

1. Jorge Colazo and Yulin Fang. Impact of license choice on open source software development activity. *Journal of the American Society for Information Science and Technology*, 60(5):997–1011, 2009.
2. Daniel Izquierdo-Cortazar, Gregorio Robles, Felipe Ortega, and Jesus M Gonzalez-Barahona. Using software archaeology to measure knowledge loss in software projects due to developer turnover. In *HICSS'09*, pages 1–10. IEEE.
3. Gregorio Robles and Jesus M Gonzalez-Barahona. Contributor turnover in libre software projects. In *Open Source Systems*, pages 273–286. Springer, 2006.
4. Gregorio Robles, Jesus M Gonzalez-Barahona, and Israel Herraiz. Evolution of the core team of developers in libre software projects. In *MSR'09*. IEEE.
5. Andreas Schilling, Sven Laumer, and Tim Weitzel. Together but apart: how spatial, temporal and cultural distances affect floss developers' project retention. In *Computers and people research*, pages 167–172. ACM, 2013.
6. Yiqing Yu, Alexander Benlian, and Thomas Hess. An empirical study of volunteer members' perceived turnover in open source software projects. In *HICSS, 2012*, pages 3396–3405. IEEE.