Volume 5, Issue 4, 2015 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Coordination and Communication Tools in Free/Open Source Software: A Metrics Evaluation

Preet Kanwal*
Department of Computer Science,
Sri Guru Gobind Singh College,
Sector 26,
Chandigarh, India

Anu Gupta
Department of Computer Science and Applications,
Panjab University,
Chandigarh, India

Ravinder Kumar Singla
Department of Computer Science and Applications,
Panjab University,
Chandigarh, India

Abstract— Free/Open Source Software (F/OSS) lack proper planning schedule of traditional software since they are developed by a community of volunteers who contribute to a project in accordance with their own skills and time schedules. Due to open nature in all aspects of F/OSS development i.e. developers, roles, time schedules, infrastructure and requirements, it is a difficult task to develop metrics for F/OSS projects. This paper proposes and validates metrics to study the impact of adoption of Coordination and Communication Tools in F/OSS projects progress. Longitudinal data for a study period of 5 year has been collected for a data set of 50 F/OSS projects. Longitudinal panel data analysis has been performed using the proposed metrics. Positive correlation has been observed between CC Tool adoption and development status of F/OSS project, whereas regression analysis results provide that an increase in number of volunteers associated with Subversion code commits and Issue addressal leads to improvement in likelihood a project to be in a higher Development Status. Individually Mailing Lists, Version Management System and Issue Tracker have been observed to be positively influencing the progress of a project. The present study contributes towards filling the gap in F/OSS literature in terms of it being a longitudinal panel study with focus on Coordination and Communications Tools.

Keywords— F/OSS; Coordination; Communication; CC Tools; Metric; SourceForge.net; SVN; SRDA; Tool Adoption; PTTAP; PTTCR

I. INTRODUCTION

Traditional closed source software development generally involves a well defined project management plan. They are developed by a team wherein members have predefined roles and perform specifically allocated tasks. In contrast, Free/Open Source Software (F/OSS) development lacks a well defined project development and management plan due to its open nature in terms of developers, roles, time schedules, infrastructure and requirements. F/OSS is an innovative development methodology brought into dominance by rise and all time availability of Internet technology [1-3]. The Coordination and Communication Tools (CC Tools) are the most important component of this form of development process. The community of volunteers uses software repositories and allied coordination and communication infrastructure for cooperative development process. Due to openness in the development process of F/OSS by decentralized team of volunteers as well as changing requirements in response to user feedback, it is important to measure the various aspects of this form of development. Measurement provides a mechanism to objectively evaluate the software development process, assess the quality of technical work products and assists in tactical decision making, with the intent of continuous improvement [4].

In this paper metrics to study the impact of CC Tools adoption in F/OSS projects' progress have been formulated and validated. The remaining sections cover: Section II - Literature Review regarding Coordination and Communication Tools and metrics, Section III - Methodology for hypothesis testing and proposed metrics, Section IV- Data Analysis and Results, Section V - Discussion and finally Section VI - Conclusion.

II. LITERATURE REVIEW

Kraut *et al.* have highlighted the importance of formal and informal communication in coordination of software development in general [5]. An Extended Information System Model comprising of eight interrelated dimensions which links quality, success, communication and contributions in F/OSS projects has been proposed which acknowledges the fundamental relevance of communication in F/OSS development [6]. The majority of development activity in F/OSS utilizes asynchronous communication which has the advantage of maintaining total record of communications [7,8,9]. Intensive communications within a team as well as the availability and use of multiple artifacts have been identified as two critical success factors for open source innovation [10]. CC Tools which can be used by F/OSS projects have been suggested [11]. Tools for effective communication, coordination and overall project management have been identified as one of the five factors contributing to the notion of *quality by development* in F/OSS [12]. Due to open nature in all aspects of F/OSS development i.e. developers, roles, time schedules, infrastructure and requirements, it is a difficult task to develop metrics for F/OSS projects. Metrics used for traditional software have been applied to study evolution of

F/OSS in terms of development [13-15] and quality evolution [16-18] using very small number of projects. Metrics and Tools have been identified as major research areas in F/OSS studies [19].

Literature on metrics to evaluate the adoption of CC Tools in relation to the progress of a project is almost non-existent. Researchers have studied the impact of a version-sensitive text editor with the rationale of making effective decisions about its subsequent usage [20]. Statistical methods have been used to find correlation between numbers of bug reports to the change in project size of three projects [21]. The ratio of email messages in public Mailing Lists to commits in Version Systems of a small set of case studies selected from Apache Software Foundation (ASF) has been used as a metric which relates the communication flow within a project with its development activity. The inference from the results show that, being an intensive metric, the ratio can be used to measure project evolution [22]. Bug count has used as a metric for explaining a framework for evaluating metrics [23]. A framework of metrics for quantitative evaluation of F/OSS quality by comparative assessment of easily accessible case study based attributes is proposed [24]. Attributes which have been considered include project lifespan, popularity, number of developers, subscribers, bug ratio, supported operating system and project status [24]. A lesser subjective approach resembling the one used in [24] utilizes IEEE 1219-98 standard to derive metrics for adaptive maintenance on F/OSS quality [25]. Researchers have studied the relation of process maturity to the success of F/OSS projects, especially where processes related to coordination and communication are concerned [26]**.** Another study with focus on tools provided by only *SourceForge.net* explores the possible benefits of CC Tools on efficiency of Open Source projects using two data sets of different sizes and sampling methods. Due to non-conclusive results obtained from the study, the researchers have acknowledged the fact that the selection of Open Source projects to study the impact of CC Tools, must be made very carefully and more extensively [27]. Stewart *et al*. have used a data set of 67 projects and survey on their administrators in a cross sectional study to establish development stage of a progress as a moderating factor on performance [28].

The various studies referred above use small sets of case studies in their work. There is a need for simple metrics based on publicly available data on F/OSS hosting sites and repositories [29]. Moreover, there is a gap in literature where longitudinal panel studies in F/OSS CC Tool adoption is concerned. Therefore a need is felt to devise metrics to evaluate adoption of CC Tools in F/OSS projects. The following sections present the methodology used to develop and evaluate a set of metrics for studying the impact of adoption of CC Tools on projects' progress in terms of development.

III. METHODOLOGY

A. Aim of Study

The present study aims to propose a *Set of Metrics* that use activity data of F/OSS projects to quantify and evaluate the effect of adoption of CC Tools on the progress of F/OSS projects.

B. Data Set

SourceForge.net is one of the leading database-driven F/OSS project hosting site with a user friendly interface which supports historic and status statistics of various user activities [30-31]. The difference in available tools and project visibility is controlled by choosing a single source to formulate the data sets [32]. In order to formulate and validate the set of proposed metrics, the four phase sampling process model [33] has been used to select a sample data set of 50 projects belonging to a broad category of system software projects from *SourceForge.net*. Projects at similar stage of development status using Subversion (SVN) as Code Repository have been chosen so that comparable archived longitudinal data for projects' progress would be available. Considering that the number of downloads represents an objective measure of user interest in the project and its success, the "Popularity" measure has been taken into account.

SourceForge.net has shared its data with *University of Notre Dame, USA* for supporting academic research. The archival data for various activities other than code commits have been accessed from *SourceForge Research Data Archive* (SRDA) implemented by *University of Notre Dame, Notre Dame, USA*, through a web-based form for executing SQL queries against the relational database [8, 34,35]. The activity related to code commits is recorded within the code repositories used by individual projects and not in the *SourceForge.net* database [32]. A F/OSS project downloaded from *SourceForge.net* itself has been used to retrieve data regarding commits from SVN Logs of the projects. The data from SVN logs as well as the results of queries executed against the SRDA have been recorded in the form of text files. A relational database structure has been designed to store the data in accordance with the requirements of the study for subsequent analysis. Java Programs with embedded SQL statements have been developed and implemented to port the data from text files into the RDBMS software. The database tables have been stored in a local repository.

C. Proposed Metrics

In F/OSS, the development and the maintenance activities are interrelated. Although trace data of F/OSS development is publicly available through various CC Tools adopted by F/OSS projects, yet it is difficult to measure various aspects related to F/OSS due to the facts:

- Volunteers are organized around the basic concept that anyone can join the team at any time in any role. There are no clear boundaries between developers, users and maintainers of F/OSS. The developer pool as well as the volunteer pool may change over time.
- The software requirements and the corresponding coordination and communication requirements of the volunteers may change over the incremental longitudinal development span.
- The volunteers may use tools of their own choice for coordination and communication regarding requirements, solutions and code development.

Keeping in view the above facts, there is a need to identify simple and appropriate metrics for analyses of various valuable longitudinal statistics collected from version control repositories and *SRDA* regarding CC Tool adoption by F/OSS projects. The following *Tool Adoption Metrics* are proposed in this research work for studying the CC Tool choices as well as rates of CC Tools adoption by F/OSS projects. The metrics can be adapted for artifacts within the Issue Tracker System, if available.

1) ***Project Total Tool Adoption Proportion (PTTAP)***: A Number of CC Tools adopted by a project $\mathbf{P_i}$ out of all the CC Tools made available by the hosting site.

$$\mathrm{PTTAP}[P_i] = \frac{P_i[n]}{N} \times 100$$

where n = Project's Tool Count and N = Total Tool Count made available by the hosting site.

2) ***Project Total Tool Change Rate (PTTCR)***: Change in the number of CC Tools adopted by a project $\mathbf{P_i}$ between two time periods.

$$\mathrm{PTTCR}[P_i] = \frac{P_i[n_{new}] - P_i[n_{old}]}{P_i[n_{old}]} \times 100$$

where n = Project's Tool Count, old = Previous time period and new = Successive time period.

The metrics have been computed and further used to analyze the effect of CC Tools as determinant of F/OSS progress in a longitudinal study.

IV. DATA ANALYSIS AND RESULTS

The metrics proposed in Section III have been applied on the various attributes collected for the data set and further used to validate the following hypothesis:

In F/OSS projects, CC Tools have a positive effect on the overall progress of a software product being developed and used.

Correlation and Regression analysis have been used to validate the hypothesis. The data has been organized in the form of longitudinal panel data where a project represents the cross-section aspect and the schema readings represent the time series aspect of the panel. Panel data models i.e. Fixed effects (FE), Random effects (RE) models [36-39] and Ordered Logit (OL) models [40] for regression have been used in this study.

A. Correlation Analysis

Table I below shows the Spearman's correlation between metric *PTTAP* and *DevStatus*. There is significant (ρ_s = 0.105, p = 0.014) but relatively small positive correlation between *PTTAP* and *DevStatus*. The resulting small value of correlation coefficient ρ_s could be due to the reason that the effect of adopting or dropping any CC Tool is not immediately recognizable on the development progress of a project. Apart from documentation and screenshot facilities, all CC Tools show positive influence on project development progress. In general, the data set supports the hypothesis.

TABLE I SPEARMAN'S CORRELATION BETWEEN CC TOOLS AND DEVELOPMENT STATUS

CC Tool (N=548)	ρ_s	p-value
PTTAP	.105[b]	.014
ToolMail	.001	.981
ToolForum	.138[a]	.001
ToolNews	.036	.405
ToolDoc	-.091[b]	.033
ToolScr	-.119[a]	.003
ToolIS	.076[c]	.074
ToolWiki	.165[a]	.000
ToolWebsite	.187[a]	.000

a - significant at the 0.01 level, b - significant at the 0.05 level, c- significant at the 0.1 level

B. Regression Analysis

The progress of a project on *SourceForge.net* is depicted as seven stage development process - Stage 1: Planning, Stage 2: Pre-Alpha, Stage 3: Alpha, Stage 4: Beta, Stage 5: Production/Stable, Stage 6: Mature, Stage 7: Inactive. The iterative process of finding volunteers, identifying solutions with corresponding coding, testing, review and release management takes place for each stage [33]. The movement of a project from lower to higher stage is considered as its development progress. Therefore, in order to quantify the progress of F/OSS project on *SourceForge.net*, *DevStatus* representing each stage is used in the analysis.

As a result of all the coordination and communication that takes place among volunteers, the development progress in terms of software code of F/OSS project is stored in SVN code repository and Issue Tracking system. Therefore, total number of code commits (*SvnRevCount*) and total number of issues closed (*ISTotalClosed*), both affect the development status of a project. Similarly, each CC Tool augments the visibility of a project among prospective users who may turn into volunteers of different types associated with the project. Therefore any additional CC Tool adopted by a project at any stage would also lead to rise in number of volunteers - code committers (*SvnDC*) and issue closers (*ISDC*). This would lead to more tasks being completed and thereby increased progress of the project.

Thus, the present study builds on Two Step Regression analysis using mediation or intervening variables in panel data [41-47]. The cause and effect relationship in the present study involves a causal chain in which the effect of causal variable $\mathbf{X_i}$ on proposed dependent variable **Y** is intervened by a single variable **I**. Two sets of regression coefficients are estimated between paths from $\mathbf{X} \rightarrow \mathbf{I}$ and $\mathbf{I} \rightarrow \mathbf{Y}$. Step 1 uses either Fixed or Random effects panel data model to generate predicted values of intervening variable **I**. In step 2, Ordered Logit Model for longitudinal panel data has been performed as dependent variable *DevStatus* is an ordered categorical variable [40]. Marginal probability effects or simply the marginal effects (ME) in Ordered Logit Model quantify the change in probability of a given event defining state of *DevStatus* due to a unit change in predictor or independent variables. Thus, the overall effect of a CC Tool's adoption or dropping on *DevStatus* can be understood by analyzing the marginal effects.

The predicted probability for a project to be in any stage of development status **(Y)** after a unit change in any independent variable $(\mathbf{X_i})$ is computed from Two Step Regression model using intervening variable **(I)** is as below:

$$\hat{\mathbf{Y}}_{\mathbf{(ME)n}} = \hat{\mathbf{Y}}_{\text{nth DevStatus}} + (\boldsymbol{\beta_I} \text{ x } (\text{ME})_{\text{nth DevStatus}})$$

where $\hat{\mathbf{Y}}_{\mathbf{(ME)n}}$ is the predicted probability with marginal effects of the nth *DevStatus*.

Taking into consideration the mediation causal chain effect explained above, a set of regressions is performed to compute the effect of overall *ToolCount (or PTTAP)* as well as an *Individual CC Tool* (X_i) over the *DevStatus* (Y) of F/OSS project. Separate cases with *SvnRevCount, ISTotalClosed*, *SvnDC* and *ISDC* as the intervening variables (I) have been computed.

TABLE II EFFECT OF PTTAP ON SVNREVCOUNT AND DEVSTATUS

Panel Regression Model and Test Statistics	**Dependent Variable**	$\boldsymbol{\beta_I}$	$\hat{\mathbf{Y}}$	**Marginal Effect**
Step 1 RE Model	SvnRevCount	-87.285[a] (0.000)	---	---
Step 2 Ordered Logit Regression	DevStatus 3	---	0.002734	5.10E-07 (0.218)
	DevStatus 4	---	0.061061	1.07E-05[b] (0.021)
	DevStatus 5	---	0.834842	5.86E-06[c] (0.088)
	DevStatus 6	---	0.098465	-1.65-05[b] (0.017)
Wald $chi^2(11)$		186.92[a] (0.000)	---	---
LR $chi^2(11)$		---	41.87[a] (0.000)	

a - significant at the 0.01 level, b - significant at the 0.05 level, c- significant at the 0.1 level

TABLE III EFFECT OF PTTAP ON ISTOTALCLOSED AND DEVELOPMENT STATUS

Panel Regression Model and Test Statistics	**Dependent Variable**	$\boldsymbol{\beta_I}$	$\hat{\mathbf{Y}}$	**Marginal Effect**
Step 1 RE Model	ISTotalClosed	5.5529[b] (0.017)	---	---
Step 2 Ordered Logit Regression	DevStatus 3	---	0.002734	-8.01e-06 (0.218)
	DevStatus 4	---	0.061061	-1.67E-04[b] (0.021)
	DevStatus 5	---	0.834842	-9.21E-

				05[c] (0.088)
	DevStatus 6	---	0.098465	2.59E-04[b] (0.017)
Wald chi^2(11)		101.07[a] (0.000)	---	---
LR chi^2(11)		---	41.87[a] (0.000)	

a - significant at the 0.01 level, b - significant at the 0.05 level, c- significant at the 0.1 level

It is observed from Table II that although *PTTAP* has negative influence on *SvnRevCount* yet the marginal effects are positive from development stage 3 to stage 5. Positive marginal effects imply that the project is more likely to be in stage 3 to stage 5 if predicted value of *SvnRevCount* is increased by one unit by virtue of change in *PTTAP*. Negative marginal effects for stage 6 imply that increase in *PTTAP* is less likely to increase the probability of a project to move to stage 6. This is due to lesser possibility of any significant increase in code size beyond stage 5 i.e. the stable stage. Table III shows that *PTTAP* has significantly positive effect on total number of issues closed given by *ISTotalClosed*. In other words, an increase in number of CC Tools adopted leads to an increase in number of issues closed. Moreover, with marginal effects shifting from negative to positive from stage 4 to stage 6 successively, the likeliness of a project to be in higher development stage for a unit change in number of issues closed gets enhanced.

TABLE IV EFFECT OF PTTAP ON SVNDC AND DEVELOPMENT STATUS

Panel Regression Model and Test Statistics	**Dependent Variable**	β_I	$\hat{Y}$	**Marginal Effect**
Step 1 RE Model	ISTotalClosed	0.08217[b] (0.014)	---	---
Step 2 Ordered Logit Regression	DevStatus 3	---	0.002734	-.0005412 (0.218)
	DevStatus 4	---	0.061061	-.0113128[b] (0.021)
	DevStatus 5	---	0.834842	-.0062249[c] (0.088)
	DevStatus 6	---	0.098465	.0175052[b] (0.017)
Wald chi^2(11)		224.05[a] (0.000)	---	---
LR chi^2(11)		---	41.87[a] (0.000)	

a - significant at the 0.01 level, b - significant at the 0.05 level, c- significant at the 0.1 level

TABLE V EFFECT OF PTTAP ON ISDC AND DEVELOPMENT STATUS

Panel Regression Model and Test Statistics	**Dependent Variable**	β_I	$\hat{Y}$	**Marginal Effect**
Step 1 RE Model	ISTotalClosed	0.09696[a] (0.000)	---	---
Step 2 Ordered Logit Regression	DevStatus 3	---	0.002734	-.0004586 (0.218)

	DevStatus 4	---	0.061061	-.0095865[b] (0.021)
	DevStatus 5	---	0.834842	-.005275[c] (0.088)
	DevStatus 6	---	0.098465	.014834[b] (0.017)
Wald chi^2(11)		175.18[a] (0.000)	---	---
LR chi^2(11)		---	41.87[a] (0.000)	

a - significant at the 0.01 level, b - significant at the 0.05 level, c- significant at the 0.1 level

It is observed from Table IV and Table V that *SvnDC* increases by a factor of 0.082 (p-value = 0.014) and *ISDC* increase by a factor of 0.097 (p-value = 0.000) for one unit increase in *PTTAP*. This clearly implies that with increase in *number of CC Tools* of a project, both the *SvnDC* and *ISDC* increase. Moreover, the coefficients of marginal effects of both predicted *SvnDC* and predicted *ISDC* on *DevStatus* show a pattern of changing from negative to positive values towards higher *DevStatus*. This increase in *SvnDC* and *ISDC* leads to probability of improvement in code functionality thereby increasing the likeliness of a project to be in a higher *DevStatus.* In other words, increase in number of adopted CC Tools leads to increase in number of distinct code committers and issue closers which in turn, have a positive effect on the development progress of the project.

The two step regression was computed for each CC Tool individually also. The summary of individual effects has been presented in Table VI. Individually adoption of only two CC Tools namely, Mailing List and Website show significant positive influence on the development progress of a project in both cases of intervening variables *SvnRevCount* and *ISTotClosed*. Since Subversion has been adopted by all the projects, its positive influence is implicit. Similarly, for case where *ISTotalClosed* is taken as intervening variable, Issue Tracker's positive influence is implicit. Except Mailing List facility, all other CC Tools have a positive effect on *SvnDC*, with *ToolWiki* (p-value = 0.035) and *ToolWebsite* (p-value = 0.016) being the significant ones. The remaining CC Tools have shown little or non significant influence on *SvnDC.* Two of the CC Tools i.e. Wiki and Website are found to be having positive influence on both *SvnDC* and *ISDC* as seen in Table VI. Website is considered to be having overall positive effect due to the fact that it is significantly positive for three of the intervening variables and marginally significantly positive for just one case. Forum is positively influencing only *ISDC*. The remaining CC Tools which have positive coefficients lead to increase in number of *SvnDC* and *ISDC* although the effect may not be significant. The marginal effects of both predicted *SvnDC* and predicted *ISDC* on *DevStatus* in case of individual CC Tools also show a pattern of changing from negative to positive values towards higher *DevStatus*. Thus, with adoption of each CC Tool having positive coefficient, the likelihood of being in a higher development status increases. Wiki and Website support hypothesis completely.

Thus, taking into consideration all four cases of intervening variables, research hypothesis could be partially validated. For code activity related parameters, the positive influence of overall adoption of CC Tools i.e. *PTTAP*, is observed for number of issues closed only and individually Mailing List and Website have been observed to be positively influencing the development progress of F/OSS projects. For metrics computed using volunteer counts, research hypothesis could be validated in the sense than any increase in volunteers would lead to increase in code related activity and in turn higher *DevStatus*. Individually, Wiki and Website are found to have positive influence on volunteers.

TABLE VI INDIVIDUAL EFFECTS OF CC TOOLS ON

Parameter	**Effect on**			
	SvnRevCount	**ISTotalClosed**	**SvnDC**	**ISDC**
ToolCount (PTTAP)	(-)[a]	(+)[b]	(+)[b]	(+)[a]
ToolMail	(+)[b]	(+)[b]	(-)	(+)
ToolForum	(-)[b]	(-)	(+)	(+)[b]
ToolNews	(+)	(+)	(+)	(+)
ToolDoc	(-)[a]	(-)	(+)	(-)
ToolScr	(-)[a]	(+)[b]	(+)	(+)
ToolIS	(-)	(+)	(+)	(+)[a]
ToolWiki	(-)[a]	(-)[a]	(+)[b]	(+)[c]
ToolWebsite	(+)[b]	(+)[b]	(+)[b]	(+)

a - significant at the 0.01 level, b - significant at the 0.05 level, c- significant at the 0.1 level

The limited effect of individual CC Tools is due to following facts:

First, the study focuses on the CC Tools offered by the hosting site only. On investigation of the specific websites of the projects, it has been observed that external mechanisms are being used by some projects in addition or in lieu of those of hosting site. These external mechanisms are separately maintained for particular CC Tools or implemented through a website. As *SourceForge.net* is a dynamic hosting site it cannot be ascertained that exactly when a project opted to use external mechanisms or changed resources. Documentation and Website has been adopted externally by most projects followed by CC Tools like blogs and social networking which are not available through *SourceForge.net.*

Secondly, volunteers involved in development of F/OSS projects are skilled personnel who are mostly the end users also. Competent volunteers can understand the development process from source code itself and the archived Mailing Lists [26]. Thus, they may not look beyond a small set of CC Tools. Moreover, the F/OSS volunteers may participate in the development process due to various factors [48]. The factors may include learning opportunities, skill improvement, job betterment or simply to be able to use the project [48-50]. This may further lead to their difference in preference of communication with the project's community members. Attaining a stable stage by a project may cause volunteers to leave the project due to very little or no scope for useful contribution.

Thirdly, over the longitudinal study period the projects were consistent in their adoption of CC Tools. It is observed from Table VII that the number of projects in which the *PTTCR* is negative i.e., projects which drop a CC Tool during the study period, is very small. Similar observation is made for positive *PTTCR*, although in year 2009 the adoption rate is highest. Most of the projects have zero *PTTCR* i.e. the overall number of CC Tools adopted by a project does not change. Thus the projects are stable in their CC Tools sustenance.

TABLE VII PROJECT COUNT WITH CHANGE IN TOOL COUNT (PTTCR)

PTTCR	**Project Count having**		
Schema	**Negative PTTCR**	**Positive PTTCR**	**Zero PTTCR**
sf0608	4	3	43
sf1208	1	2	47
sf0609	3	3	44
sf1209	3	8	39
sf0610	2	1	47
sf1210	2	5	43
sf0611	1	1	48
sf1211	0	2	48
sf0612	0	1	49
sf1212	1	6	43

V. DISCUSSION

Results have supported some of the hypothesized relationships while showing that actual pattern of effects varies across CC Tools. Whereas positive correlation has been observed between CC Tool adoption and development status of F/OSS project, regression analysis provided mixed results. The adoption of CC Tools leads to more issue being addressed which in turn positively affects the development progress. Individually Mailing Lists, Version Management System and Issue Tracker have been observed to be positively influencing the progress of a project. An increase in CC Tools adoption leads to increase in volunteers in *System Software Projects* which further led to projects' progress. Individually, not all CC Tools by virtue of their adoption, proved to have influence over the increase in code commits, code committers and issue closers. More volunteers could not result in more code nor could an increase in CC Tools attract code contributors to an already stable project.

The validation of metrics provides combined results which are similar in nature to independent results of three prior studies each with different data sets and analysis methodology [26-28]. Michlmayr's results are based on an assessment form survey of 80 randomly chosen projects to study the effect of process maturity on project success measured in terms of number of downloads [26]. Koch *et al.* has used two different sized data sets to study the effect of tools on efficiency of a project where efficiency is computed as a ratio of output to an input by the project [27]. Stewart *et al*. used a data set of 67 projects and survey on their administrators in a cross sectional study to study the effect of development stage of a project on its performance [28].

Although the results correspond to those in the above mentioned studies, yet the present study makes different research contribution. First, the parameter of interest is different in the present study. It studies the effect of CC Tool adoption on projects progress in terms of development activity which is a dissimilar parameter from project success and efficiency. Secondly, the present study is more extensive as it is a longitudinal study spread over a period of 5 years. Thirdly, it uses panel data models for analysis and hypotheses testing. Lastly, two metrics to study the effect of CC Tool adoption on project progress have been proposed and used for analysis.

The present study while reiterating the moderating role of development stage on project performance [28] takes the point further i.e. it describes the nature of effects of CC Tools due to development stage the project is in. The effects are positive for projects at lower stages of development and negative for those in advanced stages of development. Whereas Stewart *et al*. used survey on their administrators in a cross sectional study, the present study has an elaborate sample and actual activity data collected across various CC Tools in a longitudinal study with no scope for human judgmental bias. In contrast to the single time frame assessment which did not investigate the nature of relationship between process maturity and success [26], the present study tries to establish the relationship between CC Tools adoption and the subsequent effect on project progress in terms of development. Apart from present study being a longitudinal study, data set selection and size, metrics as well as the data analysis techniques applied are different from those of Koch *et al.* [27]. Yet, the similarity in results shows that the CC Tools influence the progress in terms of development status on similar lines as those of efficiency and success of F/OSS project.

The study reiterates the results of previous works with additional factor of longitudinal level using different data sets, parameters and analysis techniques as well as new set of metrics.

VI. CONCLUSIONS

Efficient coordination and communication is the most important factor in the progress of F/OSS projects. Longitudinal studies depicting the change in CC requirements and choice of CC Tools during the lifespan of F/OSS Project are lacking in the F/OSS research literature. There is also gap in literature where metrics related to studying the impact of adoption of CC Tools on the F/OSS progress is concerned. Existing literature focuses on a minimal data set with small number of case studies. The present longitudinal study while proposing and validating the CC Tool adoption metrics using panel data analysis makes significant research contribution. Moreover, the proposed metrics are general in nature and may be applicable on larger data sets involving more than one hosting site.

ACKNOWLEDGMENT

We are thankful to University of Notre Dame, USA, for providing access to *SourceForge Research Data Archive (SRDA)* for retrieving data on F/OSS projects.

REFERENCES

[1] The Free Software Definition. Available: http://www.gnu.org/philosophy/free-sw.html [Accessed: March 06, 2012].

[2] Open Source Software Definition. Available: http://www.opensource.org/docs/osd [Accessed: March 06, 2012].

[3] E. S. Raymond, “The Cathedral and the Bazaar”. *First Monday*, vol. 3, no. 3, 1998. Available: http://firstmonday.org/article/view/578/499

[4] R. S. Pressman, *Software Engineering: A Practitioner’s Approach*. *Fifth ed.* McGraw-Hill, 2001. pp. 79-83.

[5] R. E. Kraut and L. A. Streeter, “Coordination in Software Development,” *Communications of the ACM*, vol 38, no. 3, pp. 69-81, 1995.

[6] S. Fernandes, “Quality, success, communication and contribution in Open Source Software,” in *Proceedings of 5th International Workshop on Foundations and Techniques for Open Source Software Certification (OpenCert 2011)*, pp. 33-42. Available: http://opencert.iist.unu.edu/Papers/2011-paper-S2-A.pdf [Accessed: January 4, 2012].

[7] B. Lundell, B. Lings, P. J. Ågerfalk and B. Fitzgerald, “The Distributed Open Source Software Development Model: Observations on Communication, Coordination and Control,” pp. 8-10. Available: http://is2.lse.ac.uk/asp/aspecis/20060058.pdf [Accessed: 27 August 2011].

[8] Y. Gao, M. V. Antwerp, S. Christley and G. Madey, “ A Research Collaboratory for the Open Source Software Research,”in *Proceedings of First International Workshop on Emerging Trends in FLOSS Research and Development (FLOSS 2007)*. Available: http://www.nd.edu/~oss/Papers/FLOSS07.pdf [Accessed: January 24, 2012].

[9] S. Koch and J. M. Gonzalez-Barahona, “Open Source Software Engineering – The State of Research”. *First Monday*, vol. 10, no. SI-2, 2005.
Available: http://firstmonday.org/htbin/cgiwrap/bin/ojs/index.php/fm/article/view/1466/1381 [Accessed: August 09, 2010].

[10] G. Peng, “Critical Success Factors for Open-Source Innovation: The Case of Open Source Software Development,” *Issues in Information Systems*, vol X, No. 2, pp. 157-164, 2009.

[11] R. Gardler, “Essential tools for running a community-led project,”. *OSS Watch* Briefing Notes, http://www.oss-watch.ac.uk/ Available:www.oss-watch.ac.uk/resources/communitytools.xml [Accessed: January 4, 2012].

[12] S. A. Shaikh and A. Cerone, “Towards a metric for Open Source Software Quality,” in *Proceedings of the Third International Workshop on Foundations and Techniques for Open Source Software Certification (OpenCert 2009)*. *Electronic Communications of the EASST* , vol. 20, 2009.

[13] K. Johari and A. Kaur, “Effect of Software evolution on software metrics: An open source case study,” *ACM SIGSOFT Software Engineering Notes*, vol. 36, no. 5, 2011.

[14] G. Xie, J. Chen and I. Neamtiu, “Towards a Better Understanding of Software Evolution: An Empirical Study on Open Source Software,” in *Proceedings of the International Conference on Software Maintenance (ICSM)*, pp. 51-60, 2009.

[15] T. Mens, J. Fernandez-Ramil and S. Degrandsart, “The Evolution of Eclipse,” in *Proceedings of the IEEE International Conference on Software Maintenance (ICSM)*, pp. 386-395, 2008.

[16] H. Zhang and S. Kim, "Monitoring Software Quality Evolution for Defects," *IEEE Software*, vol 27, no. 4, pp. 58-64, 2010.

[17] A. Murgia, G. Concas, S. Pinna, R. Tonelli and I. Turnu, "Empirical study of software quality evolution in open source projects using agile practices'". CoRR, vol. abs/0905.3287, 2009.

[18] N. Drouin, M. Badri and F. Toure, "Analyzing Software Quality Evolution using Metrics: An Empirical Study on Open Source Software," *Journal of Software*, vol 8, no. 10, 2013.

[19] Preet Kanwal, A. Gupta and R. K. Singla, "Open Source Software Development: Exploring Research Perspectives," *Emerging Trends in Computing, Informatics, Systems Sciences, and Engineering, Lecture Notes in Electrical Engineering 151*, Springer Science+Business Media New York 2013, pp. 607- 617.

[20] D. Atkins, T. Ball, T. Graves and A. Mockus, "Using Version Control Data to Evaluate the Impact of Software Tools," in *Proceedings of the 21st International Conference on Software Engineering, May 16-22*, 1999, *Los Angeles, CA*, , pp. 324-333.

[21] L. Yu, S. Ramaswamy and A. Nail, "Using Bug Reports As a Software Quality Measure," in *Proceedings of the 16th International Conference on Information Quality (ICIQ)*, pp. 277-286 , 2011.

[22] S. Gala-P´erez, G. Robles, J. M. Gonz´alez-Barahona and I. Herraiz, "Intensive Metrics for the Study of the Evolution of Open Source Projects: Case studies from Apache Software Foundation projects," Available: http://oa.upm.es/14698/1/paper.pdf [Accessed: December 04, 2013].

[23] C. Kaner and W. P. Bond, "Software Engineering Metrics: What Do They Measure and How Do We Know?," *10th International Software Metrics Symposium (Metrics 2004)*, 2004. Available:http://testingeducation.org/a/metrics2004.pdf [Accessed:November 14, 2011].

[24] G. Polancic, R. Horvat and T. Rozman, "Comparative Assessment of Open Source Software Using Easy Accessible Data," in *Proceedings of 26th Int. Conf. Information Technology Interfaces ITI 2004*, pp. 673-678.

[25] D. Kozlov, J. Koskinen, J. Markkula and M. Sakkinen, "Evaluating the Impact of Adaptive Maintenance Process on Open Source Software Quality," in *Proceedings of First International Symposium on Empirical Software Engineering and Measurement*, pp. 186-195, 2007.

[26] M. Michlmayr, (2005) Software Process Maturity and the Success of Free Software Projects. In: Zieliński, K., Szmuc, T. (Eds.), Software Engineering: Evolution and Emerging Technologies. pp. 3–14. Available:http://www.cyrius.com/publications/michlmayr-process_maturity_success.pdf [Accessed: January 31, 2012].

[27] S. Koch, "Exploring the effects of Sourceforge.net coordination and communication tools on the efficiency of open source projects using data envelopment analysis," *Empirical Software Engineering*, 14, 2009, pp. 397- 417.

[28] K. J. Stewart and S. Gosain, "The Moderating Role of Development Stage in Free/Open Source Software Project Performance," *Software Process Improvement and Practice*, 11, 2006, pp. 177-191.

[29] V. Tiwari and R. K. Pandey, "Open Source Software and Reliability Metrics," *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 1, no. 10, 2012.

[30] SourceForge.net; http://sourceforge.net

[31] S. Christley and G. Madey, "Analysis of Activity in the Open Source Software Development Community," in *Proceedings of the 40th Hawaii International Conference on System Sciences* , 2007. Available:http://www3.nd.edu/~oss/Papers/HICSS40_final.pdf

[32] A. Gupta and R. K. Singla, "Evaluating User Participation in Defect Reporting among Free/Open Source Software Projects," *Journal of Computer Science*, vol. 4, no. 1, 2009, pp. 1387-1403.

[33] Preet Kanwal, A. Gupta and R. K. Singla," Sampling Process Model for Coordination and Communication in Free/Open Source Software Projects,". *International Journal of Computer Applications (0975 – 8887)*, vol. 98, no.8, July 2014.

[34] SRDA; http://zerlot.cse.nd.edu/

[35] M. V. Antwerp and G. Madey, "Advances in the SourceForge Research Data Archive (SRDA)", *The 4th International Conference on Open Source Systems - (WoPDaSD 2008)*, Milan, Italy, September 2008. Available: http://zerlot.cse.nd.edu/mediawiki/images/f/fd/Srda_final.pdf [Accessed on December 26, 2011].

[36] H. M. Park, "Linear Regression Models for Panel Data Using SAS, Stata, LIMDEP, and SPSS*," Working Paper*. The University Information Technology Services (UITS) Center for Statistical and Mathematical Computing, Indiana University, 2009.

[37] H. M. Park, "Practical Guides To Panel DataModeling: A Step-by-step Analysis Using Stata," *Tutorial Working Paper*. Graduate School of International Relations, International University of Japan, 2011.

[38] K. A. Bollen and J. E. Brand, "Fixed and Random Effects in Panel Data Using Structural Equations Models," *California Center for Population Research On-Line Working Paper Series*. University of California, Los Angeles. pp. 5-20, 2008.

[39] O. Torres-Reyna, "Panel Data Analysis Fixed and Random Effects using Stata (v. 4.2)", Data & Statistical Services, Princeton University. Available: http://dss.princeton.edu/training/Panel101.pdf [Accessed: 30 Oct 2014].

[40] The Econometrics Academy, Available: https://sites.google.com/site/econometricsacademy/

[41] K. J. Preacher and A. F. Hayes, "Asymptotic and resampling strategies for assessing and comparing indirect effects in multiple mediator models," *Behavior Research Methods*, vol. 40, no. 3, 2008, pp. 879-891.

[42] A. F. Hayes and K. J. Preacher, "Quantifying and Testing Indirect Effects in Simple Mediation Models When the Constituent Paths Are Nonlinear," *Multivariate Behavioral Research*, 45, 2010, pp. 627- 660.

[43] A. Gelman, "Two-Stage regression and Multilevel Modeling: A Commentary," *Political Analysis*, 13, 2005, pp. 459–461..

[44] V. P. Sipple, "Estimating engineering cost risk using logistic and multiple regression," M.Sc. Thesis, Air force Institute of Technology , Wright-Patterson Air Force Base, Ohio, 2002.

[45] J. V. Bielecki, "Estimating engineering and manufacturing development cost risk using logistic and multiple regression," M.Sc. Thesis, Air force Institute of Technology , Wright-Patterson Air Force Base, Ohio, 2003.

[46] D. C. Genest, "Logistic and multiple regression – the two-step approach to estimating cost growth," M.Sc. Thesis, Air force Institute of Technology , Wright-Patterson Air Force Base, Ohio, 2004.

[47] B. L. Lewis, "Two-stage approaches to regression models in which the dependent variable is based on estimates," July (2000), Available: http://polmeth.wustl.edu/media/Paper/lewis00.pdf?origin=publication_detail [Accessed: 25 Jan 2015].

[48] K. R. Lakhani and R. G. Wolf, "Why Hackers Do What They Do: Understanding Motivation Effort in Free/Open Source Software Projects," *Perspectives on Free and Open Source Software*, The MIT Press, Cambridge, Massachusetts, London, England, 2005. Available:http://mitpress.mit.edu/books/chapters/0262562278.pdf [Accessed: February 16, 2010].

[49] Y. Ye and K. Kishida, "Toward an Understanding of the Motivation of Open Source Software Developers," in *Proceedings of International Conference on Software Engineering (ICSE2003), Portland, Oregon, USA*, 2003. Available:http://www.informatik.uni-trier.de/~ley/db/conf/icse/icse2003.html [Accessed: March 06, 2012].

[50] A. Hars and S. Ou, "Working for Free? Motivations for participating in open source projects," International *Journal of Electronic Commerce*, vol. 6, no. 3, 2002, pp. 25-39.

