Journal of Data Acquisition and Processing
ISSN: 1004-9037
https://sjcjycl.cn/
DOI: 10.5281/zenodo.98549511

EFFECTS OF SOFTWARE DEVELOPMENT USING XP ON PRODUCTIVITY AND COST OF CHANGE

Dr. Sanjay Singh Bhadoriya

Dept. of Computer Science, Dr. APJ Abdul Kalam University-Indore

Saurabh Parikh

Research Scholar, Dr. APJ Abdul Kalam University-Indore

Abstract

Due to the complex nature of the variables at play, three distinct research strategies—a survey, a simulated case study, and formal experiments—were used to determine XP's impact on productivity. Due to a growth in development projects and a lack of established development processes, software development and effort estimates have gained substantial attention from practitioners and the industry. Planning a development project that takes into account both the current and future development environment is critical to its success. Development projects have a large amount of poor code in their code base. A number of dangers might arise if unstructured code changes are made without enough test coverage. It is necessary to gather information on both input and output parameters. Information was gathered from finished software projects by means of a survey. The information comes from 29 initiatives carried out by 24 firms in a variety of nations. An issue that often arises in multiple regression analysis is the presence of a strong relationship between the input variables.

Keywords: development, Software Problem Report, XP, Productivity, Cost of Change

INTRODUCTION

In the beginning of the development process, the reporting and definition of problems is done. Problem reports are used as a means of quoting change requests during development. Software Problem Reports (SPRs) are provided by users of the system in present techniques of development. Reporters may specify the level of importance and urgency of a certain issue. As soon as an essential element of the program is missing, it's deemed vital. Unless the issue report specifically indicates a timeframe for a remedy, it is prioritized by the software review board. Development receives all SPR and diagnoses them. A software change request (SCR) is created by the Development team to address a specific issue. During the meeting of the software review board, SPRs and related SCRs are brought up. An SPR is either selected or rejected by the board. Sequential code and requirement details must be updated on the SCR form. Key inputs for project bidding, budget, personnel allocation, project planning and control etc. are derived from these documents. Project failures may be attributed to unrealistic and incorrect estimations. The software engineering community has worked hard to build models that assist estimators in generating precise costs estimates for software projects, realizing the significance

of software estimating. COCOMO, SLIM, SEER-SEM, and Parametric Review of Information for Costing and Evaluation-Software Estimating Model are just a few of the numerous software estimate models that have been created and utilized during the past three decades (PRICE- S). Rather of starting from scratch, the software industry has opted to improve current systems rather than start from scratch. As a consequence of patched and re-patched software, legacy projects often include unstructured code that has to be addressed by developers.

LITERATURE REVIEW

Mitre-Hernandez, Hugo & Bermon, Leonardo (2014) In the software development business today, agile practices are gradually displacing more conventional process-based approaches. Nonetheless, process-based software development is still significant because of the amount of code that can be reused in subsequent projects. When Knowledge Management (KM) is not properly linked with processes, issues like low output and stagnant development might result. This chapter provides a case study of implementing two process library recommendations in an agile software development team. To discover examples of tacit and explicit knowledge that affect process improvement and productivity, researchers and the software development division used a Kanban dashboard, eXtreme Programming (XP), and SCRUM practices, adapted to the Process Assets Library (PAL) and the Process Practice Library (PPL), across three software projects. In "results and discussion," the writers lay out both successful and unsuccessful strategies.

Bermón-Angarita, Leonardo (2014) The software industry is now undergoing a shift from old process-based methodologies to agile ones. Nonetheless, process-based software development is still significant because of the amount of code that can be reused in subsequent projects. Lack of productivity and opportunities for process improvement might result from a lack of alignment between Knowledge Management (KM) and processes. Two concepts of process libraries are implemented in an agile software development department and their results are presented here as a case study. Using a Kanban dashboard, eXtreme Programming (XP), and SCRUM practices modified for the Process Assets Library (PAL) and the Process Practice Library (PPL) in three software projects, researchers and the software development division sought to identify examples of tacit and explicit knowledge with an effect on process improvement and productivity. The authors detail both successful and unsuccessful procedures in the results and comments section.

Tavakoli, Farzaneh & Javdani Gandomani, Taghi (2018) Software development teams have always focused on improving team productivity. Team productivity is a major concern in the highly competitive software business since the software development process is a team-oriented process. Several research have been done on different elements of team productivity, but due to the importance of the methodology used, it is impossible to provide a cohesive model. This article's goal is to present a productivity model tailored to the XP technique with an emphasis on increasing team efficiency. The most important factors influencing team productivity served as the basis for the suggested model. There was sufficient evaluation and

acceptance of this model. The best XP techniques that have a direct and beneficial influence on team productivity are also highlighted in this article.

Matharu, Gurpreet & Mishra, Anju & Singh, Harmeet & Upadhyay, Priyanka (2015) The software business has become increasingly sophisticated as a result of technology advancements and the changing needs of customers. As agile guarantees early software development and high-quality software products, it is being embraced by software practitioners. It also allows for rapid adaptation to new user needs as they emerge throughout the development process. In this article, we discuss the importance, advantages, and growing popularity of Agile techniques in light of the complex and ever-changing requirements placed on software today. The primary goal of this research is to empirically investigate the decision-making process behind selecting one of the three most common Agile methodologies: Scrum, Extreme Programming, or Kanban. In addition, this article offers a comparison of many agile software development approaches. According to the findings of our poll, Scrum-based development is more popular in the modern software business than Extreme Programming and Kanban.

Forsgren, et al (2021) Developer productivity is not only about how much work a developer puts in or how well the engineering methods used to release software perform. Here, we show how the SPACE concept can be used to real-world productivity analysis, and we argue that doing so will help teams better understand developer productivity and provide more insightful metrics with which to guide their work and teams.

RESEARCH METHODOLOGY

Research Model

Separating the effects of XP techniques on productivity from those of other variables is a crucial step for the study model. The model should also be able to measure how much of impact XP strategies have on output. In this case, the statistical models perform better. They are also simple to interpret, which is why a statistical model based on multiple regression was chosen for this study. To examine the correlation between many independent predictors and a single outcome, statisticians use the multiple regression model. The multiple regression model uses observed data to infer the connection between input and output variables. The non-linear multiple regression equation based on power functions is chosen from a variety of available alternatives.

$$P = A \cdot X_1^{B1} \cdot X_2^{B2} \ldots X_k^{Bk}$$

Where P is productivity, A is constant, and X_1, X_2 X_3… X_n reflect the contributors to productivity. B_1, B_2,B_3…B_n are coefficients in a regression model indicating the relative importance of different variables in predicting production.

Data Collection

It is necessary to gather information on both input and output parameters. Information was gathered from finished software projects by means of a survey. The information comes from 29 initiatives carried out by 24 firms in a variety of nations. Information from uncompleted tasks was disregarded. Due to the confidentiality agreement with the firms, the specifics of the data are not provided in this.

- The parameters of the research model were estimated using the data. Regression coefficients and model fitness parameters were calculated as part of this process. Three criteria namely R^2 Adj-R^2 and the F-value is a statistical measure of how well a given model fits the data.

- Analyzing the EXP parameters allowed us to confirm the study's hypotheses.

- Estimation interval analysis was used to quantify productivity growth of b_{EXP}.

Test for Multicollinearity

An issue that often arises in multiple regression analysis is the presence of a strong relationship between the input variables. The term for this is multicollinearity. A possible issue with multicollinearity exists if a correlation coefficient matrix including all input variables reveals correlations of 0.75 or higher. In the event of multicollinearity being found, the more strongly linked (more than or equal to 75%) of the two variables should be eliminated. Hence. Our regression model requires us to investigate the possibility of multicollinearity among the input variables. Research model input variables (PRODUCT, DEYT, ENVR, PROCES.9, and UP) were tested for multicollinearity using the gathered data.

DATA ANALYSIS

Effects of XP on Software Development Productivity

Model Fitness

The goal of a regression model is to make predictions about an output variable by testing the hypothesis that its value would be similar to the mean value obtained from a data set of observations. Three criteria namely R^2, Adj-R^2 How well a model fits the data may be evaluated using the F-value.

- R^2 represents the fraction of the total variation that can be attributed to the variables chosen for the analysis. R^2 is called the coefficient of determination, which might be anywhere from zero to one. An R^2 When the neatness of a model reaches 1.0, it explains almost all of the variation in the output variable, and adding more input variables is not expected to make the model better.

- Adj-R^2 is since it takes into account the total number of model parameters, it provides a more accurate measure of the "goodness of fit". Adjusted R^2 is always less than R^2.

- An F-test evaluates the overall statistical significance of the regression equation.

Calculated and provided are the model fitness parameters in table 1.

Table 1: Fitness parameters of the model

R^2	Adj-R^2	Std. Err.	F-value	P-value for the model
81.46%	78.37%	0.105	26.37	0.0

Table 1 reveals an R^2 of 82% for the aforementioned model. This means that input factors can explain 82% of the output variable's variability. This means that the model is appropriate. Because the F-test p-value is zero, we may conclude that there is a substantial correlation between the input and output variables.

Effects of XP on Cost of Change

The findings suggest that XP approaches have a substantial impact on software development output. There is a minimum 54% gain in productivity and a maximum 173.8% increase in productivity when the adoption levels of all XP practices are raised from the ratings Not Followed (Ratings RI) to Almost Always (Ratings R6). This finding is consistent with other studies finding a productivity boost of 66% (Poole and Huisman) and 67% (Maurer and Martel). Note that not all of these procedures have been used in previous case studies. It's also worth noting that certain techniques are implemented in part. When Planning Come is at the Nor Followed setting, the box plots show that all but one of the data points fall inside the normal distribution. While this data point may seem out of the ordinary at first glance, it has been included in the study since no anomalies have been found. Table 2 offer descriptive statistics for these data, respectively.

Table 2: Descriptive Statistics of output data

	Rang	Minimum	Maximum	Mean	Std. Dev	Coeff. of. Var.
All Practices are followed as recommended	22.94	75.41	98.35	89.16	6.97	7.82
Absence of on-site customer	62.25	80.87	143.13	117.49	16.91	14.39
Absence of Refactoring	78.16	75.77	153.94	116.78	28.26	19.91

Absence of Small Releases	63.69	77.62	141.31	116.57	18.61	15.96
Absence of planning Game	49.64	77.62	127.26	112.15	13.95	12.44
Absence of Simple Design	62.31	77.33	139.64	110.14	18.99	17.24
Absence of collective Code Ownership	51.98	78.51	130.50	106.83	15.90	14.88
Absence of Pair Programming	5457	78.40	132.98	104.52	17.92	17.15
Absence of Metaphor	48.49	77.11	126.60	104.09	15.51	14.90
Absence of Coding Standard	52.67	77.80	130.48	103.28	17.51	16.95
Absence of Forty-Hours Week	50.12	77.55	127.67	102.74	15.44	15.03
Absence of Continuous Testing	50.20	77.62	127.83	101.34	15.90	15.69
Absence of Continuous integration	33.86	75.96	109.93	92.99	10.01	10.76

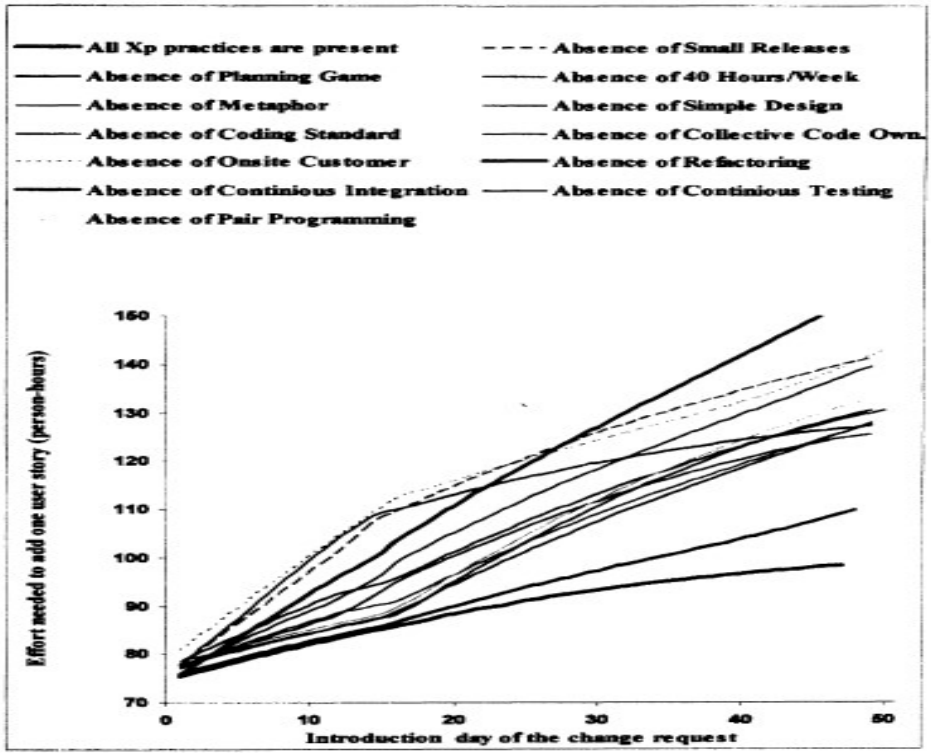

Figure 1: Cost of change curve when a specific practice is set to 'Not followed'

Twenty-nine software projects' worth of data shows that XP practices adoption (EXP) significantly impacted software development productivity. Once the impacts of other variables on productivity have been accounted for, an improvement in productivity of between 54% and 173.8% may be achieved by raising the adoption level of all XP practices from the lowest level (Not Adopted) to the greatest level (Almost Always). In other words, productivity increased by 9.1% to 22.3% for every tier up in the adoption of XP methods. Therefore, XP principles do allow software organizations to boost efficiency during software development. Using the model, we were able to determine how various XP techniques affected the overall price of transitions.

CONCLUSION

A system dynamics simulation model of a real-world XP project (a case study) was built and used to examine the impact of XP on the price of change. The cost of change curve was calculated by repeatedly simulating the model. Using the model, we were able to determine how various XP techniques affected the overall price of transitions. Formal tests were done to evaluate the efficiency of learning when pair programming is adopted with that of the conventional approach (solo programming) for completing short-duration programming assignments in order to determine the impacts of pair programming on learning efficiency.

REFERENCES

1. Mitre-Hernandez, Hugo & Bermon, Leonardo. Process and Productivity Improvement in Agile Software Development with Process Libraries: Case Study, 2014, 10.4018/978-1-4666-5182-1.ch001.

2. Bermón-Angarita, Leonardo. Process and Productivity Improvement in Agile Software Development with Process Libraries: Case Study, 2014

3. Tavakoli, Farzaneh & Javdani Gandomani, Taghi. A Novel Team Productivity Model for XP Teams. Journal of Cases on Information Technology. 20. 93-109, 2018, 10.4018/JCIT.2018100106.

4. Matharu, Gurpreet & Mishra, Anju & Singh, Harmeet & Upadhyay, Priyanka. Empirical Study of Agile Software Development Methodologies. ACM SIGSOFT Software Engineering Notes. 40. 1-6, (2015). 10.1145/2693208.2693233.

5. Forsgren, Nicole & Storey, Margaret-Anne & Maddila, Chandra & Zimmermann, Thomas & Houck, Brian & Butler, Jenna. The SPACE of Developer Productivity: There's more to it than you think. Queue. 19. 20-48, (2021). 10.1145/3454122.3454124.

6. Darwish, N. R. Enhancements in Scum Framework Using Extreme Programming Practices. International Journal of Intelligent Computing and Information Sciences (IJICIS), Ain Shams University, 14(2), 53-67, 2014

7. Malhotra, C., & Chug, A. IXSCRUM-A Framework Combining Scrum and XP. International Journal of Scientific & Engineering Research, 4(7), 2013

8. Rasool, G., Aftab, S., Hussain, S., &Streitferdt, D. (2013). eXRUP: A Hybrid Software Development Model for Small to Medium Scale Projects. Journal of Software Engineering and Applications, 6(09), 446, 2013

9. Ashraf, S., & Aftab, S. Latest Transformations in Scrum: A State of the Art Review. International Journal of Modern Education and Computer Science (IJMECS), 9(7), 12-22, 2017

10. Alshehri, S., &Benedicenti, L. Prioritizing CRC cards as a simple design tool in extreme programming. In Electrical and Computer Engineering (CCECE), 2013 26th Annual IEEE Canadian Conference on (pp. 1-4). IEEE, 2013