Quantitative Process Improvement in XP Using Six Sigma Tools

Sajid Ibrahim Hashmi and Jongmoon Baik
Information and Communications University,
109 Munji-ro, Yuseong-gu, Daejon,
South Korea
{hashmi,jbaik}@icu.ac.kr

Abstract

XP (eXtreme Programming) has been used for reducing defects and dealing with changing customer requirements effectively. It has been well known for its defect reduction capability. But, software can be made more reliable by means of process improvement. In general, models like CMM/CMMI have been used for software development process improvement. However, these models have some problems to be used with XP because of less number of process artifacts produced by agile processes. Six Sigma on the other hand, provides the quantitative analysis capabilities required to measure and control process performance. Using XP with Six Sigma can provide means of analyzing XP data and systematically improve process performance. In order to prove our point, we map Six Sigma tools to activities involved at each phase of XP and show that Six Sigma technique can be used with XP to measure and improve the performance of XP process. We also suggest a Six Sigma process improvement guideline with examples.

1. Introduction

Software defect reduction has been the ultimate goal of practitioners and researchers. The journey started with the advent of CMM[1] then CMMI[2], meanwhile various other process improvement techniques were also introduced; e.g. GQM[3]. The ultimate goal of these techniques is to tune the organizational process so that defects can be minimized and eventually customer satisfaction can be achieved [1][2][3]. One of the major causes of defects is considered to be inability to convey and understand the requirements both by customers and developers respectively. To meet this challenge, agile processes have been introduced, which are flexible enough to accommodate requirements at any stage of software development life cycle [4]. Since there is no requirement sign off and customer is available on site almost all the time, requirements can be conveyed and better understood to developers. Historically, it has been difficult to sail the ferry of agile processes in the ocean of these capability maturity models for three reasons: 1) different activities in agile processes are being carried out at the same time so it is difficult to identify the practices at each phase. 2) CMM/CMMI requires large number of process artifact data for different phases of any software development life cycle, which is not produced by agile processes. 3) CMM/CMMI focus on big organizational structure whereas agile processes are mostly followed in parts of a big organization or small organizations for the development of certain components, with small number of developers working on it.

XP is considered to be the most famous agile process [5]. This popularity is earned by its various practices like refactoring, system metaphor, simple design, and TDD (Test Driven Development). These practices which themselves work for reducing defects and thus build quality in the process [4]. Six Sigma, on the other hand, provides the quantitative analysis tools, which are useful to control process performance. A Six Sigma project can improve a business characteristics by 70 % or more, simulating increased operating margins for business, while at the same time increasing the value to their customer [6]. It provides a suite of different tools like correlation analysis, scatter plots, affinity diagrams, ANOVA, FMEA, and t-test, which can be well used for measurement with the minimum set of required data. Using Six Sigma with XP can enable to systematically improve process performance by analyzing XP process data. Continuing with this rationale, we identify practices involved at each phase of XP and map Six Sigma tools to these practices in order to show that Six Sigma technique can be applied to XP process data analysis. We also suggest Six Sigma practical guideline to be used with XP.

978-0-7695-3131-1/08 $25.00 © 2008 IEEE
DOI 10.1109/ICIS.2008.53

Authorized licensed use limited to: Korea Advanced Institute of Science and Technology. Downloaded on May 12,2010 at 00:44:50 UTC from IEEE Xplore. Restrictions apply.

Rest of the paper is organized as follows: sections two and three give introduction to XP and Six Sigma respectively, section four describes how Six Sigma can be used with XP, and finally section five gives summary of the research.

2. eXtreme Programming

An XP release starts with Exploration phase in which business and developer discuss what they should do. At first place this phase facilitates assessing the expected functionality; technically developers can make a tradeoff between existing and proposed technology. Planning game determines the scope of next release. The development team estimates the effort and technical risks associated with implementation of each story, and approximates the number of hours required to spend in a given iteration. Productionizing phase ensures that developers must start development of elicited stories in the form of iterations. Various XP practices are being used at this stage like system metaphor, solo or pair programming, simple design, and TDD.

After the initial set of stories is implemented, Maintenance phase can accommodate new stories by customers, or even modifications in the previously implemented stories. XP practices like continuous refactoring and integration are applied here to make sure that despite of story interruptions by customer, the

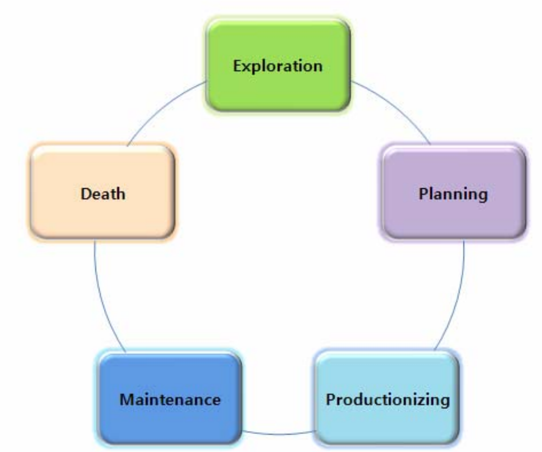

Fig 1. XP Phases

overall iteration system is in working order all the time. Death is the end of iteration. Small documentation is produced which lists the expected and actual artifact values and customer satisfaction level achieved by the end of that iteration. This written documentation is a measure of value delivered by an XP iteration.

In any agile process, it is difficult to identify activities involved at each phase because these are going on almost at the same time [4]. Table 1 shows the XP phases, corresponding practices, and the role played by each of the practice. It is notable that most of

Table 1. XP Phases

Phases	Practices	Description
Exploration	On Site Customer Story Elicitation	Problem definition, Knowledge integration, and to display the current level of story understanding
Planning	Planning Game System Metaphor	To estimate new iteration size and development time
Productionizing	System Metaphor Pair Programming Test Driven Development Simple Design Continuous Integration Sustainable pace Coding standard	To analyze cause and effect relationship in terms of productivity or number of defects
Maintenance	On Site Customer Refactoring Continuous Integration Collective Code Ownership Design Improvement	To calculate the variance in productivity or quality across different system iterations
Death	On site customer	To verify if planned and actual values are similar and within the calculated process limits

Authorized licensed use limited to: Korea Advanced Institute of Science and Technology. Downloaded on May 12,2010 at 00:44:50 UTC from IEEE Xplore. Restrictions apply.

the practices are repeated but still certain practices are specific for certain phases; e.g. onsite customer is compulsory for Exploration phase for story elicitation and planning game; development of system metaphor is necessary for Planning phase; pair or solo programming, simple design, coding standard, and continuous integration are essential for Productionizing phase; collective code ownership, refactoring, and design improvement are important for Maintenance phase; finally onsite customer is required to confirm that no more stories should be implemented in that specific iteration. The good thing about XP practices is that they all complement each other. Planning game and system metaphor are developed based on stories elicited by onsite customer; refactoring is carried out by continuously modifying and integrating the existing design so that further stories or changes can be accommodated; and finally the iteration release ensures that all practices have been followed properly to satisfy the customer.

3. Six Sigma

Six Sigma is a problem solving methodology and a statistical term for a process which improves quality by decreasing the number of defects [7]. A Six Sigma organization can use Six Sigma methodology to improve performance, lower cycle time, reduce complexity, minimize defects and errors, and increase customer satisfaction. It minimizes mistakes and maximizes value by means of measurement. It has several classes of tools and technology to accomplish its goal:

- Tools for designing, modeling, managing, and optimizing processes.
- Tools for broad scale management of multiple projects across multiple organizational units.
- Tools for collecting data, conducting analytical calculations, and solving performance problems.
- Tools and technologies for training, educating, transferring and managing knowledge.

Six Sigma does not have fixed number of tools and also these tools may be named differently [8]. Some specific tools which support team development can be used more efficiently if pair programming is used. The good thing about Six Sigma is that it uses a consistent approach for experimentation [7]:

- Accumulates information about process or system.
- Provides insight knowledge about different variable interactions in a process.
- Quantifies the amount of knowledge discovered about a system.

4. XP and Six Sigma

4.1. Six Sigma Toolset for XP

For selecting six sigma tools for XP, process elements can be chosen in many ways; for example process measures and estimation models. Once the elements are identified, Six Sigma tools that provide required functionalities for analyzing the elements are included in a set of Six Sigma tools for XP. For instance, defects data collected throughout XP life cycle need to be prioritized to find defects type to focus on, which will then be used to update design or coding standard to prevent the defects from reoccurring in future. Also, productivity and estimation measures must also be prioritized for reuse purpose.

Some of the Six Sigma tools for XP are team based tools [8]. The applicability of such tools is justified more if we use pair programming practice of XP. The identified team based tools are affinity diagrams, SQFD, Kano analysis, and SWFMEA. Moreover, process mapping seems to have less applicability to XP since there is no complex process to be mapped in case of agile processes. We identified the set of Six Sigma tools for XP; e.g. pareto analysis, cause and effect diagram, control charts, ANOVA, t-test, scatter plots, correlation analysis, and regression analysis.

The relationship between Six Sigma tools for XP process artifacts should be defined for use of these tools for process improvement. For this reason, we must define Six Sigma tools required to apply at identified XP activities. In other words, it can be said that these activities provide input necessary to apply Six Sigma tools. In XP, activities are divided into different phases of process. Since it is possible for each practice to be carried out at each phase, same Six Sigma tools can be used across each phase to control and improve these practices.

4.2. Examples of Using Six Sigma Tools with XP

As claimed earlier, Six Sigma tools can be used with XP. In order to prove the point, we use these tools to measure some of the process artifacts produced by XP:

4.2.1. Duration Estimates

Decision about number and duration of each iteration is made during Planning phase. The idea is to manage resources before Production phase so that

Authorized licensed use limited to: Korea Advanced Institute of Science and Technology. Downloaded on May 12,2010 at 00:44:50 UTC from IEEE Xplore. Restrictions apply.

development goals can be achieved on short term basis. The estimation is normally based upon two things: 1) historical story data, 2) amount and nature of current stories. Accuracy knowledge of historical estimates may assist engineers and practitioners to better forecast their future iterations and projects estimates. Normally these estimates are made in terms of working hours or weeks. Table 3 [9] shows the estimated and actual hours across different iterations of an XP project.

Table 2. Estimated and Actual Durations

Iteration No	Estimated Hours	Actual Hours
1	60	55
2	60	61
3	60	63
4	70	65
5	70	66
6	100	99
7	100	102

Student's *t*-test [10] is used to test the hypothesis we developed with our case study results. This test can be used to compare the means of two distributions. Here we have distributions in the form of actual and estimated hours.

Table 3. t-Test for Variance

	Estimated	Actual
$\bar{X}$ Mean(total / n)	74.29	73
σ^2 Variance	346.70	366.34
σ_d Standard deviation	10.10 $=\sqrt{\sigma_d^2}$ (the standard deviation of the difference between the means).	
$t = (\bar{X}_1 - \bar{X}_2 / \sigma_d)$	0.13	

By entering the t-table [11] at 12 degrees of freedom ($n_1 + n_2 - 2$), we get 2.18 (normally $p = 0.05$) as the tabulated value of t. Our calculated t value is well below the tabulated value which means that difference between the two means is lowly significant. It indicates that estimated and calculated iteration duration estimates are almost the same. The analysis for *t*-test always considers variance and it is valid only if variances of iterations/cycles are similar. There is a simple test to check if two variances are equal in statistical terms: divide the larger variance by the smaller (366.34/346.70=1.06) and compare the resultant variance ratio with a value from table of 'F' [12] for $p=0.05$. For two treatments, there is one degree of freedom between them. The tabulated F value is 4.8. Our variance ratio (1.06) is less than this. It means performed *t*-test was valid and both variances do not differ significantly.

To study whether an observed input has effect on observed output, we create a set of box and whisker plots of the critical output, with each box and whisker plot corresponding to a different condition of input variable. Potential sources of variation can be identified through graphical analysis. Graphically we can easily notice difference between centers of variation between two variables. Variance of two replicates can also be checked using simple variance test. We use Minitab [13] for this purpose. Both F-Test and Levene Test can be used for this kind of investigation. We have to establish two kinds of hypothesis before we proceed:

H_0 (Null Hypothesis) : Two means are equal
H_1 (Alternative Hypothesis) : Two means are not Equal

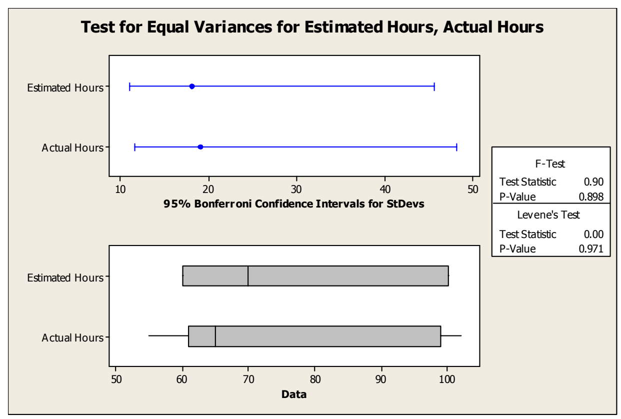

Figure 2. Variance Test for Estimated & Actual Hours

As shown in Figure 2; values calculated by both tests are 0.89 and 0.97, which are greater than 0.05. It implies that null hypothesis cannot be rejected, that both means are equal (there is no significant difference between them).

Previously, we have demonstrated the accuracy check for data values within a single project. In addition, there may be situations when estimations have to be made using historical data using the accuracy of estimates across numerous historical projects. Regression analysis can be used to check the validity of previously available data. Figure 3 is a regression plot of duration estimates of different XP projects. The data values were sampled using BOOTSTRAP [14] methodology.

Authorized licensed use limited to: Korea Advanced Institute of Science and Technology. Downloaded on May 12,2010 at 00:44:50 UTC from IEEE Xplore. Restrictions apply.

Figure 3. Regression Plot between Estimated & Actual Values

It is apparent from Figure 3 that there is positive relationship between variables and there is not much variance difference either. Figure 4 is summary statistics of regression analysis.

```
The regression equation is
Actual Hours = 69.2 + 0.057 Estimated
Hours

Predictor          Coef     SE Coef
T      P
Constant           69.23     19.15
3.62   0.002
Estimated Hours    0.0566    0.2572
0.22   0.828

S = 6.62382, R-Sq = 0.3%,    R-Sq(adj)
= 0.0%

Analysis of Variance

Source            DF      SS       MS
F       P
Regression         1    2.12     2.12
0.05   0.828
Residual Error    18    789.75   43.88
Total             19    791.87
```

Figure 4. Summary Statistics of Regression Analysis

As shown in the Figure, predicted hours values are used as predictors; whereas, actual hours are used as response. Coefficient value of "Estimated Hours" (0.0566) is significant and cannot be rejected. It indicates that for each 1 hour increase in estimated hours, the percentage of actual hours is expected to increase by 0.056%. Furthermore, regression results tell that predictor is significant because of its low P-value. Also, R-Sq value suggests that predictor accounts for 0.3% of the variance of actual hours. It is clear from Figure 4 that null hypothesis cannot be rejected since the value of P is 0.828 which is well above 0.05.

4.2.2. Iteration Duration Effect on Productivity

Duration estimates for each iteration are made during the Planning phase. Normally these decisions are made independent of customer. Since productivity is a big concern of XP, having close look at relation between iteration duration and amount of productivity achieved, can better allow engineers and practitioners to revise their forecasting for future. Table 4 contains duration measures for five iterations of an XP project with corresponding productivity achieved in terms of LOC/h.

Table 4. Iteration Duration with Productivity

Iteration No	Duration (weeks)	Productivity (LOC/h)
1	2	13.39
2	2	25.12
3	2	16.63
4	1	9.02
5	1	20.05

Figure 5 is a graphical representation of the relation between productivity achieved across five iterations of one and two weeks of length. The notable thing is that data obtained from iterations of same lengths is quite variable. From the statistic results (Figure 6) and Figure 5, the positive relation between iteration duration and LOC/h produced can be observed.

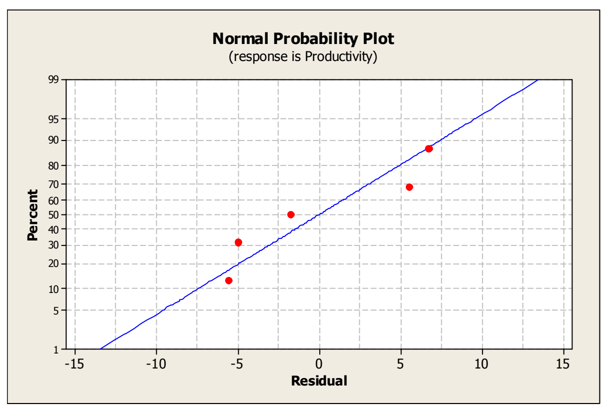

Figure 5. Normal Probability Plot between Iteration Duration vs. Productivity

Authorized licensed use limited to: Korea Advanced Institute of Science and Technology. Downloaded on May 12,2010 at 00:44:50 UTC from IEEE Xplore. Restrictions apply.

```
One-way ANOVA: Productivity versus Iteration
Duration

Source   DF      SS     MS      F
P
time      1    17.7   17.7   0.40
0.574
Error     3   134.2   44.7
Total     4   152.0

S = 6.689,R-Sq = 11.67%,R-Sq(adj)=
0.00%

Level   N     Mean   StDev
1       2   14.535   7.799
2       3   18.380   6.058
```

Figure 6. Summary Statistics of Regression Analysis

In the above figure, time (No of weeks) is considered to be the source of variation. DF is degrees of freedom from each source, which is equal to 1; SS is sum of squares between groups (factor); and error is the sum of squares within groups. MS (mean squares) are obtained by dividing the sum of squares by degrees of freedom. F is calculated by dividing the factor MS by the error MS. Value of p (0.574) is above 0.05 which indicates that the factor is insignificant, meaning that iteration duration doesn't have significant effect over productivity.

5. Discussion

In this paper, we proposed process improvement guideline for XP using Six Sigma tools. We first identified the XP practices at each phase of process and then identified set of Six Sigma tools to be used with these practices. Using Six Sigma with XP can further improve the estimation and accuracy and customer satisfaction of XP products by minimizing the defects and meeting the project schedule.

The agile nature makes it difficult to predict the detailed schedule. These less detailed artifacts and schedule produced by XP must also be controlled for defects, productivity, and planned vs. actual results for further improvement. A key principle of XP is to focus on program code and neglect dependencies among programming tasks. On the other hand, the philosophy of Six Sigma is data analysis by means of measurement, which is key to any kind of improvement. Different Six Sigma tools can be used to analyze the same kind of process artifacts in different ways, e.g. variance difference in two distributions can be checked differently using t-Test, Variance Test, ANOVA, and Regression Analysis. The obtained information can be used to improve the process by analyzing the measurement results.

6. Acknowledgement

This research was jointly sponsored by Higher Education Commission of Pakistan and MIC(Ministry of Information and Communication), Korea, under the ITRC(Information Technology Research Center) support program supervised by the IITA(Institute of Information Technology Advancement)" (IITA-2008-(C1090-0801-0032)).

7. References

[1] http://www.sei.cmu.edu/cmm/

[2]CMMI version 1.1, Continuous Representation, tech. report CMU/SEI-2002-TR-011, Carnegie Mellon Univ., 2002.

[3]Y. Mashiko and V. Basili, "Using the GQM Paradigm to Investigate Influential Factors for Software Process Improvement," Journal of Systems and Software, vol. 36(1): 17-32, January 1997.

[4]Sajid Ibrahim Hashmi and Jongmoon Baik, "Software Quality Assurance in XP and Spiral: A Comparative Study", International Conference on Computational Science and its Applications, ICCSA 2007, IEEE Page(s): 367-374.

[5]M. Angioni, D. Carboni, S. Pinna, R. Sanna, N. Serra, A. Soro, "Integrating XP Project Management in Development Environments", Journal of System Architecture, 2006.

[6]Thomas Pyzdek, "The Six Sigma Handbook", McGraw-Hill, 2001.

[7]Craig Gygi, Neil DeCarlo, Bruce Williams, "Six Sigma for Dummies", WILEY, 2005.

[8]Youngkyu Park, Hyuncheol Park, Hojin Choi, Jongmoon Baik, "A Study on the Application of Six Sigma Tools to PSP/TSP for process Improvement", IEEE/ACIS ICIS-COMSAR 2006, Page(s):174 – 179.

[9]Aftab Ahmed, Muhammad Moazam Fraz, Fahd Ali Zahid, "Some Results of Experimentation with Extreme Programming Paradigm", 7th International Multi Topic Conference, 2003. INMIC 2003. IEEE Page(s):387-390

[10]http://helios.bto.ed.ac.uk/bto/statistics/tress4a.html

[11]http://helios.bto.ed.ac.uk/bto/statistics/table1.html

[12]http://helios.bto.ed.ac.uk/bto/statistics/table3.html.

[13] www.minitab.com

[14]Michael R. Chernick, "Bootstrap Methods : A Practitioner's Guide", WILEY, 1999.

Authorized licensed use limited to: Korea Advanced Institute of Science and Technology. Downloaded on May 12,2010 at 00:44:50 UTC from IEEE Xplore. Restrictions apply.