

A Six Sigma Approach for Software Process Improvements and its Implementation

Punitha Jayaraman, Kamalanathan Kannabiran, and S.A.Vasantha Kumar.

Abstract— Six Sigma is a data-driven leadership approach using specific tools and methodologies that lead to fact-based decision making. This paper deals with the application of the Six Sigma methodology in reducing defects in Maintenance Projects of a Software industry. The DMAIC (Define–Measure–Analyze–Improve–Control) approach has been followed here to solve the underlying problem of reducing the customer reported defects in user acceptance testing phase of the software development lifecycle. This paper explores how a Software process can use a systematic methodology to move towards world-class quality level.

Keywords— DMAIC , Six Sigma, Software industry, Quality.

I. INTRODUCTION

MOTIVATED by the theme that product quality is determined by the process which produces the product [3], process improvements have become an attractive area to improve the product quality. Six Sigma was developed based on this theme to reduce the variance in the processes. Thereby, process and product quality improvement can be achieved. With the success stories of adopting Six Sigma and achieving high ROI (Return On investment) in manufacturing organizations like Motorola, Allied Signal and General Electric [1], many software organizations have tried to adopt Six Sigma and initiated Six Sigma projects to improve their software development processes from requirements, to design, implementation, and testing continuously with an ultimate goal of high customer satisfaction with high quality products.

However, Six Sigma adoption in software industry is different from the ones in traditional manufacturing industry because of the intangibility, complexity, and changeability of software products. Six Sigma principles as an effective methodology in software industry, stress on reducing variation and eliminating the root causes of defects. In contrast to the other improvement methodologies Six Sigma metrics and tools

are able to measure the defects rate, analyze the performance and improve the quality level in software projects. This paper dispels the myths concerning the unsuitability of Six Sigma in the software arena. At the same time it highlights the status of Six Sigma implementation in a software organization and the best practices for implementation of Six Sigma. As an example adopting Six Sigma principles to decrease the customer reported defects in Maintenance projects has been explained in this report.

II. BACKGROUND

In this section, some background information about Six Sigma and DMAIC methodology are introduced.

A. Six Sigma

Six Sigma is an effective and systematic quality improvement approach to enhance the organization's performance based on the adoption of various statistical analytic techniques [2]. The primary goal of Six Sigma is to reduce the variances in the processes by eliminating defects that interfere with customer satisfaction, and reducing the cost on the organization's development processes. Six Sigma has been conceived as the managerial strategy for quality improvement by quantitatively evaluating organization's processes and reducing process variances [4]. Six Sigma is described in terms of three perspectives [5]:

- **Philosophy:** Being more profitable, Six Sigma can be used for improving customer satisfaction by eliminating defects.
- **Metrics:** As a metric, Six Sigma means 3.4 DPMO (Defects Per Million Opportunities). Additionally Six Sigma includes several metrics such as Defect rate (Parts Per Million), Sigma Level, DPU (Defects per unit), and Yield [6].
- **Improvement framework:** Six Sigma owns various toolkits and structured problem solving methodologies such as DMAIC and DFSS (Design For Six Sigma)

B. Six Sigma Methodology: DMAIC

A typical Six Sigma methodology for the existing process improvements has 5 phases: Define, Measure, Analyze, Improve, and Control. DMAIC methodology can be used to find problems in existing processes and fix them for improvements. It can also be used to expand the current

Punitha Jayaraman is a M.Tech student in Department of Industrial Engineering & Management, DSCE, Bangalore, Karnataka-560078, India. (phone: +91 -9886549395 ; e-mail: jayaramanpunitha@gmail.com).

Kamalanathan Kannabiran, is a Programme Director , Quality Enabling Function Team, MindtreePvt.Ltd,Bengaluru-560059,Karnataka, India. (e-mail: Kamalanathan_Kannabiran@mindtree.com).

Dr.S.A.V.Vasantha Kumar, is a Professor in Department of Industrial Engineering & Management, DSCE, Bangalore-78, India.(e-mail: vasantha_k@yahoo.com).

capabilities of an existing process by identifying opportunities to improve current processes. Each phase of DMAIC is explained as follows:

- *Define* phase is to define project goals aligned with business goals, project scope, customers with their requirements, project charter and project teams. A high-level map of the current process is also created.
- *Measure* phase is to collect data about current processes, and develop measurement systems to validate collected data. Based on measured data, the current process performance is calculated.
- *Analyze* phase is to identify ways to decrease the gap between the current performance level and the desired goals. The project team analyzes collected data of current processes, and determines the root causes of the poor sigma performance of the processes.
- *Improve* phase is to identify, evaluate, and select the right improvement solutions. Focusing on the root causes identified in Analyze phase, the project team generates and selects a set of solutions to improve sigma performance.
- *Control* phase is to implement the final solutions and guarantee the maintenance of newly improved processes so that the improved sigma performance holds up over time.

C. Six Sigma project on Process improvement

This section discusses a Six Sigma project on the improvement of the software process, which will give a better understanding of the approach, methodology and benefits of Six Sigma.

II. DEFINE PHASE

The define phase is summarized by the project charter for the reduction of customer reported defects(Fig 1).The charter results from several meetings with the project team. The project black belt, process owner, and local champion are the key team members who formulate the project charter. These meetings provide the means for identifying the business problem and revising it until the final version of the project charter results.

III. MEASURE PHASE

In the Define Phase our Problem Statement focused on the reduction of the Customer reported defects so that the overall Organizational performance efficiency could be improved. We need to collect data to analyze to evaluate the hypothesis in the Define Phase. A high level SIPOC (Table I) tool is constructed to identify all the relevant elements of the process.

Project Charter				
Problem Statement: During April 2012 to January 2013 the Customer Reported Defects in the Maintenance Projects had accounted for 14% which was 9% higher than the Organizational Objective.				
Goal Statement : Reducing Customer Defect rate from 14% to 5% by two Quarters.				
Project Scope & Boundaries: In Scope: Customer related Metrics of Maintenance Projects of Travel and Media Business Unit of the organization. Out of Scope: All non- Maintenance Projects.				
CTQ: Customer Defect rate.				
Timeline: Completion of control phase by May 1				
Deliverables: Revised process to improve efficiency and reduce the Customer reported defects in the Maintenance Projects. Documentation and training will be provided for a revised system. Other changes will be assessed at the end of the analyze phase.				

Fig.1 Finalized Project Charter

Table I SIPOC Analysis

Suppliers	Input	Processes	Output	Customers
1.Project Managers 2.Delivery Managers 3.Quality Function	1.MIS Report 2.Radar 3.Empower 4.MS Excel Sheets	See Fig (3)	1.Bar chart 2.Pareto chart 3.Run chart	1.Delivery Team 2.Quality Team

A. Data collection table

The Data collection plan included the team to collect the data on the Report Maintenance Enhancement Metrics and to focus on the number of customer reported defects from the MIS Report. The Data validated (Table I) that the User Acceptance Testing Defects were higher in the Customer reported defects.

Table II Phase wise Customer Reported Defects

S.NO	Phase	Number of Defects
1.	Review Analysis	0
2.	Testing	47
3.	User acceptance testing	90
4.	Post Release	18

Source : MIS Report for Maintenance projects

We also collected data on the defects of software development life cycle of the project. The data indicated (Table II) most of the defects were identified only on the Coding and Testing phases i.e. at the later stages of the life cycle.

Table III Phase wise SDLC Defects

S.NO	Phase	Number of defects
1.	Requirement Analysis	385
2.	Designing	32
3.	Coding	1236
4.	Integrated testing & Software testing	934

Source: MIS Report for Maintenance projects

B. Sigma calculation

The project team had the problem of higher customer reported defects in their Maintenance Projects. Based on the metrics of the Past 10 months data from the MIS report, the long-term process capability sigma(Lt) was at 0.69σ , while the short-term capability sigma(St) was at 2.19σ , and the Defects Per Million Opportunities (DPMO) were at 245225. A Six Sigma approach was initiated to improve the quality of deliverables. The goal of the project was to improve the sigma value from 2.19σ to more than 3σ

III. ANALYZE PHASE

Process performance was assessed using Cause-and-Effect diagrams, to isolate key problem areas, to study the causes for the deviation from ideal performance, and to identify if there is a relationship between the variables. A Hypothesis testing using chi-square test was done and the probability value obtained was less than 0.05, ($p=0.00 < 0.05$) which validated that the UAT defects and the increase in customer reported defects are related. Pareto analysis Technique (Fig. 4) was also employed to analyze the defects, it says that more than 80% of the defects are injected in User Acceptance Testing phase. Extensive brain-storming sessions were also held with team members to evolve a Cause & Effect diagram. Fig 5 shows the Cause-and-Effect diagram for the piloted project showing higher customer reported defects.

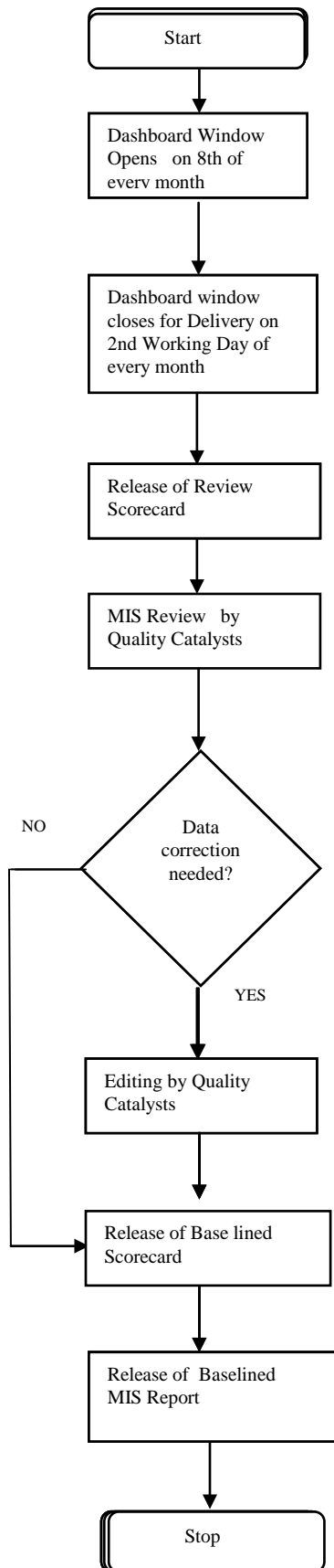


Fig.3 Process Map

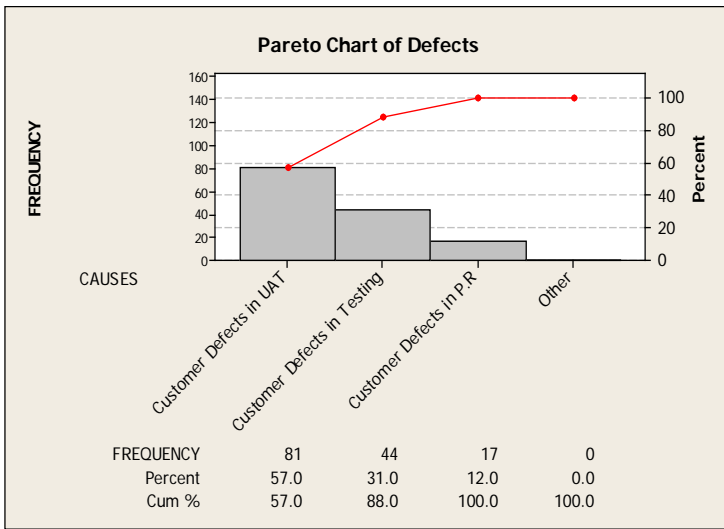


Fig.4 Pareto Analysis

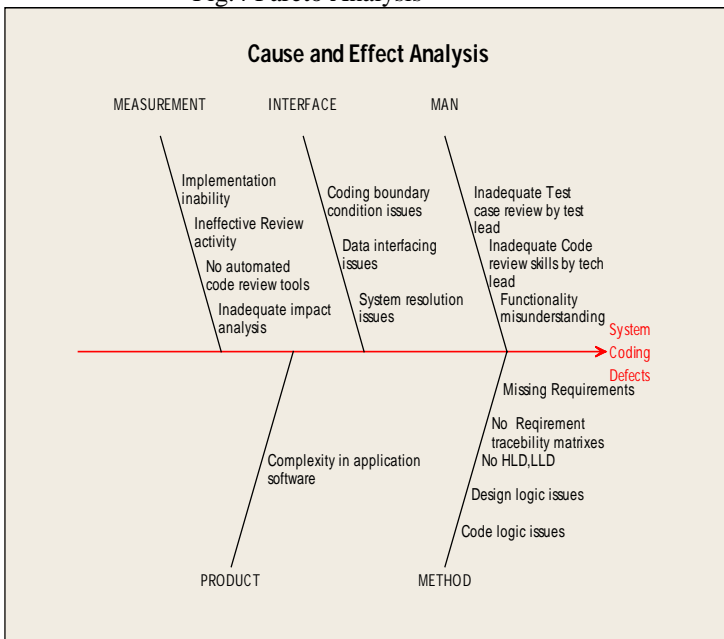


Fig.5 Cause and Effect Analysis

The probable causes that can lead to increase in Customer reported defects in a project during different phases of a project life cycle were listed. The Failure Modes and Effects Analysis (FMEA) was subsequently carried out (Table III) to arrive at a plan for prevention of causes for failure. FMEA is a tool that helps prevent the occurrence of problems by identifying the potential failure modes in which a process or product may fail to meet specifications, and rating the severity of the effect on the customer, providing an objective evaluation of the occurrence of causes, determining the ability of the current system to detect when those causes or failure modes will occur. Based on the above factors, a Risk Priority Number (RPN) for each failure mode is calculated.

IV. IMPROVE PHASE

Based on the FMEA analysis the following recommendations are followed and it resulted in several process improvements in the Requirement, Coding and Testing phases. Few critical aspects are provided below

- Creation of functional specification in parallel with the development of the code as it brings clarity from the perspective of developer and the tester.
- Requirement Traceability should be automated. Change Request and Requirement Traceability log should be from the same source
- Review all the regression test cases (have a *Test Type of Regression Test Cases*). If there are any that are no longer applicable to the regression test case set, then change it as outdated
- Self-review and Peer review of the code helps reduce the defects related to algorithm implementations, incorrect logic or certain missing conditions.

Table III FMEA Analysis

Potential Failure Modes	Potential Causes	OCC	SEV	DET	Risk priority Number
Incorrect/ Incomplete Test cases	User Interface issues w.r.t front end and Backend	7	9	7	441
Code Documentation Defects	Understanding difficulty regarding the functionality of the code between Developer and the Tester.	9	9	5	405
Missing Requirements	Improper mapping of from requirement phase to testing phase	7	8	6	336
Algorithmic and Processing Defects.	Induces new defects because of the current fixes	6	9	4	216

V. CONTROL PHASE

The control phase is responsible to ensure the sustainability and development of the improvements that have been obtained. Improvements that were introduced in one of the project team resulted in the reduction of field errors. Process capability for quality of deliverables improved from 1.3σ to 2.0σ . Control charts (FMR) were drawn to track the process level (process characteristic within projects) and process variation (process characteristic between projects) simultaneously, and also to detect the presence of special causes. Fig. 6 shows the improvement for one of the project team before and after the recommendations being followed.

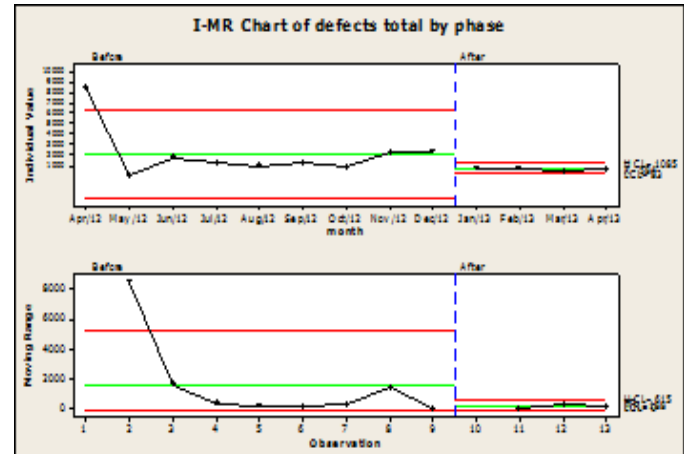


Fig.6 I-MR Chart for piloted project

REFERENCES

- [1] Cvetan Redzic and Jongmoon Baik, "Six Sigma Approach in Software Quality Improvement", International Conference on Software Engineering Research, Management and Applications (SERA'06), August 2006.
- [2] Thomas Pyzdek, "The Six Sigma Handbook : A Complete Guide for Green Belts, Black Belts, and Managers at All Levels", McGraw-Hill, 2003.
- [3] W. S. Humphrey, Managing the Software Process, Addison- Wesley, Reading, MA, 1989
- [4] Pete, Larry Holpp, What is Six Sigma?, McGraw-Hill, 2002
- [5] Jeannine Sivi, M. Lynn Penn, and Erin Harper, "Relationships Between CMMIPP P@PP P and Six Sigma", Software Engineering Institute, CMU/SEI-2005-TN-005, 2005
- [6] David L. Hallowell, "Six Sigma Software Metrics, Part 3", <http://software.isixsigma.com/library/content/c031015a.asp>