International Journal of Mathematical Archive-4(9), 2013, 258-263
Available online through www.ijma.info ISSN 2229 – 5046

SIX SIGMA IMPLEMENTATION IN SOFTWARE COMPANIES USING DTS

Anuradha[1], D. S. Rao[2*], Anil Khandekar[3], Titu Singh Arora[4] and Gurleen Kour[5]

[1,2,3,4]Computer science & engineering Dept. IIST, Indore, (M.P.), India

[5]Assistant Professor, Mahant Bachittar Singh College of Engineering and Technology, Jammu (J&K), India

(Received on: 25-07-13; Revised & Accepted on: 20-08-13)

ABSTRACT

Six Sigma has been very successfully applied in manufacturing. Is it also applicable to software, and ICT? Six Sigma is about cost reduction by eliminating defects. Six Sigma uses the following three principles: Focus on customers, Process orientation, Leadership based on metrics The big question as to whether Six Sigma can really be applied as successfully in the software industry as it was to manufacturing is still being debated. The real challenge is to see if it can be implemented without reinventing the wheel. There is also disagreement among leaders in the software industry about the need for Six Sigma. Applying Six Sigma to software development makes software projects transparent to both management and customers. Transparency requires an important cultural change. As a result, after transparency is achieved, completing accurate project estimations while meeting both deadlines and customer requirements becomes a lot easier. I had implemented DTS as a prototype by using techniques of six sigma to track an error. With the help of graph I also show the no. of bug incremented or decremented versus date of track.

Key Terms- *DTS, Six Sigma.*

I. INTRODUCTION-THE SIX SIGMA APPROACH

Six Sigma [10] is about measuring defects in the value chain process in order to systematically reduce them and therefore the corresponding cost factors. Sigma (represented as σ) is a statistical term that measures the deviation from the target set. A process that executes on 6σ level will yield results within the tolerance interval with 99.99966% probability.

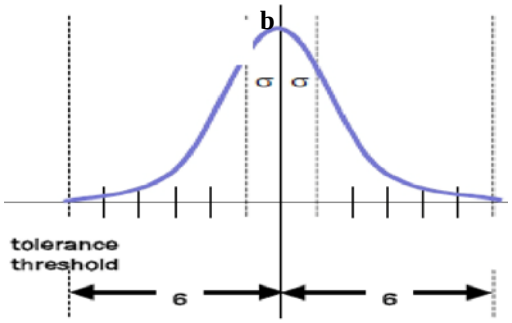

Figure 1. Six Sigma Measure

Thus 6σ corresponds to only 3.4 defects per 1 million defect opportunities. A defect opportunity is a measurable process result that is critical to customer satisfaction. Six Sigma is not team driven like TQM. It is a management approach. A defect is only what affects customer satisfaction. Thus before you can start Six Sigma you need to know what your customer really needs. Six Sigma is process – oriented. The term "customer" refers to who is using the output of your process and this is not always the end– customer. It is thus important to define what "defect" exactly refers to, when using a Six Sigma approach.

Corresponding author: D. S. Rao[2]*
[2]Computer science & engineering Dept. IIST, Indore, (M.P.), India

II SIX SIGMA AS A METRIC, METHODOLOGY, MANAGEMENT SYSTEM

Six Sigma has been labeled as a metric, a methodology, and now, a management system. While Green Belts, Black Belts, Master Black Belts, Champions and Sponsors have all had training on Six Sigma as a metric and as a methodology, few have had exposure to Six Sigma as an overall management system. Reviewing the metric and the methodology will help create a context for beginning to understand Six Sigma as a management system. [13]

A. Six Sigma as a Metric

Sigma is the measurement used to assess process performance and the results of improvement efforts - a way to measure quality. Businesses use sigma to measure quality because it is a standard that reflects the degree of control over any process to meet the standard of performance established for that process. Sigma is a universal scale. It is a scale like a yardstick measuring inches, a balance measuring ounces, or a thermometer measuring temperature. Universal scales like temperature, weight, and length allow us to compare very dissimilar objects. The sigma scale allows us to compare very different business processes in terms of the capability of the process to stay within the quality limits established for that process. The Sigma scale measures Defects Per Million Opportunities (DPMO). Six Sigma equates to 3.4 defects per million opportunities. The Sigma metric allows dissimilar processes to be compared in terms of the number of defects generated by the process in one million opportunities.

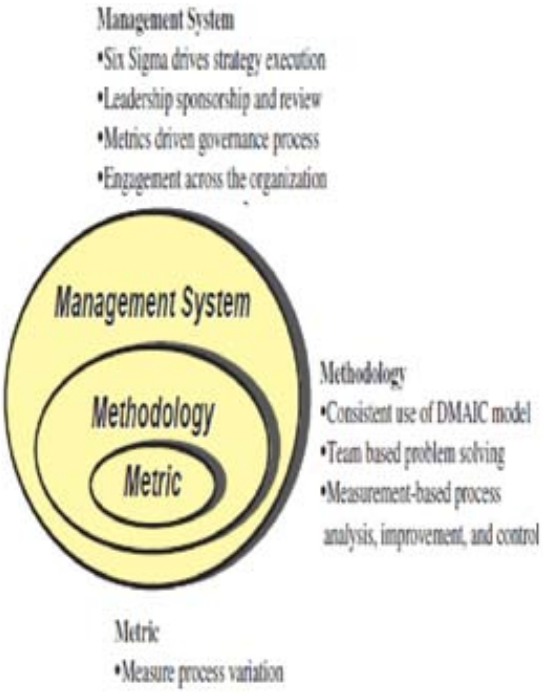

Figure 2. Six Sigma as a Metric, Methodology, Management System

B. Six Sigma as a Methodology

The Six Sigma methodology builds on the Six Sigma metric. Six Sigma practitioners measure and assess process performance using DPMO and sigma. [16] They apply the rigorous DMAIC (Define, Measure, Analyze, Improve, Control) methodology to analyze processes in order to root out sources of unacceptable variation, and develop alternatives to eliminate or reduce errors and variation. Once improvements are implemented, controls are put in place to ensure sustained results. Using this DMAIC methodology has netted many organizations significant improvements in product and service quality and profitability over the last several years.

C. Six Sigma as a Management System

Six Sigma as a best practice is more than a set of metric-based problem solving and process improvement tools. At the highest level, Six Sigma has been developed into a practical management system for continuous business improvement that focuses management and the organization on four key areas:

- understanding and managing customer requirements
- aligning key processes to achieve those requirements
- utilizing rigorous data analysis to understand and minimize variation in key processes
- driving rapid and sustainable improvement to the business processes.

III. BENEFITS USING SIX SIGMA

In manufacturing industry, hidden costs because of waste and rework in production processes are a major opportunity for cost savings. In Information and Communication Industry (ICT) this is even truer. When using every day's Office applications, everybody knows how often the many defects in the software affect our productivity and effectiveness. For ICT projects, defect costs are even more important. There exist various estimations that identify hidden cost due to defects in ICT, depending upon the industry and application area. Some Benefits are Teambuilding, Constant improvement, Mandated training, Return on investment

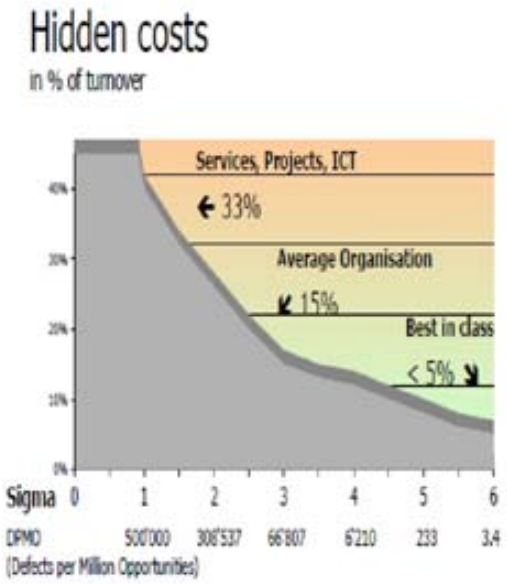

Figure 3. Hidden Costs

The famous CHAOS Report of the Standish Group [12] noted repeatedly that only about 20% of all ICT projects reach their targets. The rest will either be stopped (~30%) or miss important requirements such as functionality or deadlines (~50%). This means that we have an opportunity to get the benefits of ICT for significantly less cost, up to 30% or better. Indeed, organizations that use such an approach consistently report cost saving or margin improvements of this size**.**

IV. DOES IT WORK FOR SOFTWARE?

Nevertheless, there were only occasionally attempts to implement Six Sigma for software. Applying Six Sigma to software development makes product development and other projects transparent to both management and customers. However, transparency requires an important cultural change. We know from experience that bad communication is a major reason why projects fail and software projects in particular. We expect from better transparency that meeting both deadlines and customer requirements becomes easier. The major problem with early Six Sigma attempts was that there was no connection of software metrics to economic success [6].

Counting mistakes and defects is not a clear indication if the software project is going to be successful (e.g. [4]). Other metrics like time-to-market and user friendliness are much more important in many application areas. Sometimes, reliability is of essence, but not always. There is no software metrics that serves all. It depends what kind of software we are developing or installing. Finding good metrics for software development or deployment is a major task in itself.

V. PROPOSED ARCHITECTURE

The major problem with early Six Sigma attempts was that there was no connection of software metrics to economic success [6]. Counting mistakes and defects is not a clear indication if the software project is going to be successful (e.g. [4]). The second problem with six sigma is the communication gap between programmer and tester. A deliverable is a uniquely defined and validated work result. The project consists of a sequence of deliverables. Measuring the degree of completion for the deliverables that are needed for the project goals makes it possible to understand the project's progress as a whole. We can measure time line, cost and quality targets. Every deliverable must go through six steps.

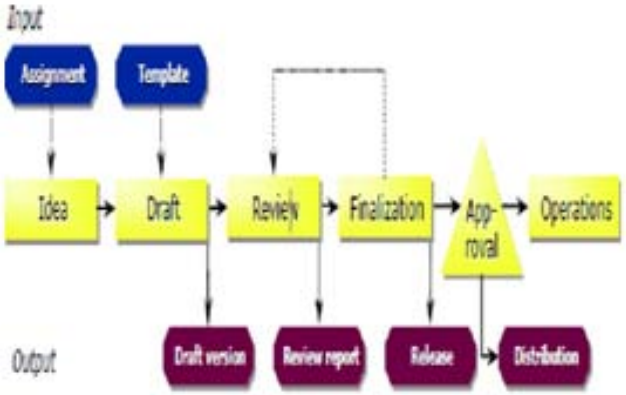

Figure 4. Six steps to completition

Among these are draft, review, and finalization before approval and usage. It is possible to get objective evidence to identify the six development steps by recording the first draft version, the review findings, the final version, and the distribution list. Even for complex projects, such an overview is extremely helpful.

Each of the six steps needs a percentage of the total duration. We set 10% for the idea, 30% for the draft, 15% for review, 20% for finalization, 15% to get approval, and 10% to make it operational. This progress grid yields progress metrics that can be accumulated for every project.

We calculate a progress index that indicates overall progress and compare with planned and actual resource consumption. This metric enables the responsible project manager to assess whether he can meet the schedule commitments, and thus identifies the measures. The assessment is based on objective evidence for progress, namely the planned quality assurance activities that must go with the deliverable.

Quality records from reviews and tests provide objective evidence for the completion of a task, if the task has a deliverable. Pending item lists are no longer needed to cope with unfinished tasks [9]. When using an integrated development with a document management repository, the state of the deliverable is monitored automatically. GMC's Czech development office has integrated all their task management in *Bugzilla* [2], an open source environment. Reports and progress metrics are produced automatically, without bothering the developers. Planned schedules are now met regularly, without pain. The defect count is how many deadlines committed to customers are missed.

VI. PROPOSED SOLUTION

I had design a prototype named as DTS by using techniques of six sigma to track the errors. It also reduces the gap between the programmer and tester. I had developed this prototype in NET framework. I had also add testing such as how many total critical defect, how many functional defect, how many cosmetic defect I will also try how to:

- Achieve transparency
- Understand the six steps for managing software development projects and listen to an explanation of how it impacts software development, software testing, project management, and project controlling
- Significant reduction in defects.
- Development time is less.
- Try to Reducing cost.
- To minimize the gap between programmer and tester.
- Review of the final product makes easier.
- Improved reliability and predictability of software products and services.

Feature of our Tool:

- No data duplication
- No Paper Work Required
- Time Efficient
- Cost Efficient
- Automatic data validation
- User friendly environment
- Data security and reliability
- Fast data insertion & retrieval
- Easy performance check
- Maintaining records of customers
- Crystal Report generation

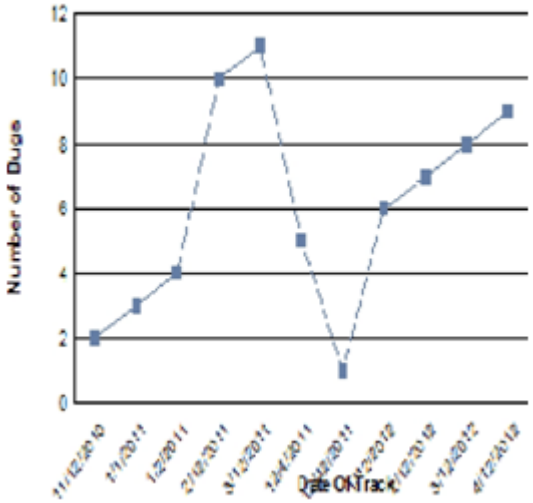

Figure 5. Defect tracked Graph (No. of bugs versus Date of Track)

Implementation includes a graph between no. of bugs and date of track as shown in fig. As soon as no. of bugs increment than according to date of track the graph will goes up and goes down according to no. of bugs reduce.

VII. CONCLUSION

Six Sigma is about cost reduction by eliminating defects. Applying Six Sigma to software development makes software projects transparent to both management and customers.

The big question as to whether Six Sigma can really be applied as successfully in the software industry as it was to manufacturing is still being debated. The real challenge is to see if it can be implemented without reinventing the wheel. What would be the tester approach while testing the application? How will the tester be responsible for improving the processes in the company? I had implemented a prototype named as DTS(Defect tracking system) that reduces the gap between the programmer and tester and also overcomes the challenges of Six Sigma. With the help of this tool we can find severity defect such as high, medium, low severity defect and priority defect such as low, medium, high priority defect. A graph is generated no. of bugs versus date of track. Advantage of our project are security, performance, efficency, control.

REFERENCES

[1]Akao Yoji. et.al: Quality Function Deployment (QFD); Productivity Press 1990, Portland, OR

[2] Bugzilla Project Home Page

[3] Cohen, Lou: Quality Function Deployment. How to Make QFD Work for You, Prentice Hall PTR 1995, New Jersey, NJ

[4] Fenton, Norman; Krause, Paul; Neil, Martin: A Probabilistic Model for Software Defect Prediction, IEEE Transactions on Software Engineering

[5] Fehlmann, Thomas: Risk Exposure Measurements on Web Sites, in: 4th European Conference on Software Measurement and IT Control, FESMA, May 2001, Heidelberg, Germany

[6] Fehlmann, Thomas: Business oriented testing in e-Commerce, in: Software Quality and Software Testing in Internet Times, April 2002, SQS AG, Köln, Germany

[7] Fehlmann, Thomas: Combinatory Metrics for Software Development, in: 8th International Symposium in QFD, September 2002, Munich, Germany

[8] Fehlmann, Thomas: Strategic management by business metrics: an application of combinatory metrics, International Journal of Quality and Reliability Management, Vol. 20, No 1, 2003 pp.134–145.

[9]Fehlmann, Thomas: Metrics for Project Management, in: 14th Annual UK Software Metrics Association Tutorials and Conference, August 2003, Wolverhampton, UK.

[10] Fehlmann, Thomas: Linear Algebra for QFD Combinators, in: 9th International Symposium on QFD, December 2003, Orlando, FL

[11] Snee, Ronald: Leading Six Sigma: A Step-by-Step Guide Based on Experience with GE and Other Six Sigma Companies, Financial Times Prentice Hall 2002

[12] The Standish Group: CHAOS Chronicles v3.0, 1994 – 2003, http://www.standishgroup.com/chaos/toc.php

[12] Humphrey, Watts: Managing the Software Process, Addison Wesley Longman, Reading 1989, MA

Source of support: Nil, Conflict of interest: None Declared

