DOI:26524/jms.2017.23

Journal of Management and Science ISSN: 2249-1260 | e-ISSN: 2250-1819

.

A methodology to improve software and information technology business using six sigma

MS.U.URATHAL ALIAS SRI SWATHIGA,
Assistant Professor, Department of Computer Science,
DRBCCC Hindu College, Pattabiram, Chennai-72.

ABSTRACT

Six Sigma is a method that provides organizations tools to improve the capability of their business processes. In six sigma a set of management techniques intended to improve business processes by greatly reducing the probability that an error or defect will occur. This increase in performance and decrease in process variation lead to defect reduction and improvement in profits, employee morale, and quality of products or services.

A software company is needed a reliable measurement data to make a case for process improvement. Unfortunately that data normally comes from an improvement project. In this paper we use Six Sigma to Building a Business Case for Software Defect Reduction.

Keywords: Six Sigma, statistical tool, Defect Reduction, Improve Business Process.

INTRODUCTION

Statistics is the mathematical science involving the collection, analysis and interpretation of data. A number of specialties have evolved to apply statistical theory and methods to various disciplines. Six **Sigma** is a disciplined, data-driven approach and methodology for eliminating defects in any process – from manufacturing to transactional and from product to service.

Six sigma is a set of techniques and tools for process improvement. It seeks to improve process quality through minimizing variability and the identification and error removal causes. It is the systematic use of statistics. Using a set of process assumptions, a six sigma quality level is usually defined as 3.4 defective parts (errors) per million opportunities.

Six Sigma approach is a collection of managerial and statistical concept and techniques that focuses on **reducing variation** in processes and preventing deficiencies in product. Standard Deviation is used to measure variance. The concept of Variation states **–NO two items will be perfectly identical."** A six sigma process is one in which 99.9999966**% of** the products manufactured are statistically expected to be free of defects (**3.4 defects per million**).

A dabbawala is a person in India, most commonly in Mumbai, who is part of a delivery system that collects hot food in lunch boxes from the residences of workers in the late morning, delivers the lunches to the workplace, predominantly using bicycles and the railway trains, and returns the empty boxes to the worker's residence that afternoon.

6σ

Each Six Sigma project carried out within an organization follows a defined sequence of steps and has specific value targets.

2. LIST OF INDUSTRIES USING 6 σ:

- Aerospace and Defense
- Airline
- Automotive
- Business Services
- Chemicals
- Computers & Electronics
- Construction
- Consumer Food Products
- Consumer Products
- Energy
- Financial Services
- Forest & Paper Products
- Government/Non-Profit
- Healthcare
- Hospitality
- Industrial
- Insurance
- Law Enforcement
- Military
- Pharmaceuticals
- Restaurants
- Software/IT
- Transportation & Logisitics
- Travel & Leisure

- Financial – bank of America, GE Capital,HDFC,HSBC,American Express
- ITES- ICICI One source, Accenture, Satyam PO, IBM Daksh
- Hospitality- ITC Hotels, GRT Hotels, Apollo Hospitals
- Manufacturing- GE Plastic, Johanson and Johnson, Motorola, Nokia, Microsoft, Ford, Wipro, Nestle, Samsung, Samtle
- Telecom- Bharti Cellular, Vodafone, Siemencs,Tata
- IT- Wipro, Satyam,Acenture, Infosys, TCS, Birla soft

3. Six Sigma METHODOLOGIES:

Six Sigma projects follow two project methodologies. These methodologies composed of five phases.

1 DMAIC is used for projects aimed at improving an existing business process.
2 DMADV is used for projects aimed at creating new product or process designs.

3.1 DMAIC:

The DMAIC project methodology has five phases:

- ***Define*** the system, the voice of the customer and their requirements, and the project goals, specifically.

- ***Measure*** key aspects of the current process and collect relevant data; calculate the 'as-is' Process Capability.

- ***Analyze*** the data to investigate and verify cause-and-effect relationships. Determine what the relationships are, and attempt to ensure that all factors have been considered.

- ***Improve*** or optimize the current process based upon data analysis using techniques such as design of experiments, mistake proofing, and standard work to create a new, future state process. Set up pilot runs to establish process capability.

- ***Control*** the future state process to ensure that any deviations from the target are corrected before they result in defects. Implement control systems such as statistical process control, production boards, visual workplaces, and continuously monitor the process. This process is repeated until the desired quality level is obtained.

3.2 DMADV

The DMADV project methodology, known as DFSS ("**D**esign **F**or **S**ix **S**igma"),features five phases:

- ***D****efine* design goals that are consistent with customer demands and the enterprise strategy.
- ***M****easure* and identify CTQs (characteristics that are **C**ritical **T**o **Q**uality), measure product capabilities, production process capability, and measure risks.
- ***A****nalyze* to develop and design alternatives
- ***D****esign* an improved alternative, best suited per analysis in the previous step
- ***V****erify* the design, set up pilot runs, implement the production process and hand it over to the process owner(s).

4. THE FIELDS USING SIX SIGMA IN SOFTWARE & IT:

(i) Building a Business Case for Software Defect Reduction for a software company to make a case for process improvement, reliable measurement data is needed. Unfortunately that data normally comes from an improvement project.

(ii) Trimming IT Fat with Lean Assessments By mapping the entire software application development life cycle, IT departments can see dramatic improvements in efficiency.

(iii) The Five Essentials for Software Testing Five essential elements are required for successful software testing: test strategy, testing plan, test cases, test data and a test environment. If any one of these five elements is missing or inadequate, your test effort will most likely fall far short of what you cold otherwise achieve.

(iv) Usage-based Models Improve Odds in Software Testing While software testing is

challenging because certain properties present difficulties, one test approach, often called "usage-based," has statistical roots and connections that Lean Six Sigma Black Belts could effectively support and perhaps evolve.

(v) Implementing Six Sigma Methodology in a Small IT Firm Six Sigma can help improve any size IT organization by providing a competitive edge. The main thing that prevents many small IT businesses from exploring Six Sigma does not know just how they can implement the data-driven methodology.

(vi) Improved IT Project Forecasting Through Six Sigma Variation in IT project cost forecasting can result in either under spent or overspent budgets. A Six Sigma approach applied to the IT project forecasting process minimizes variation in forecast accuracy resulting in dollar savings for the business.

Improving the Value of the IT Service Delivery Process

While most information technology (IT) organizations are constantly struggling to Demonstrate cost savings, too few are focusing on ways to add value to the business. Alignment with the business is where the emphasis should be for IT departments.

5. Business Case for Software Defect Reduction

One of the many challenges faced when attempting to build a business case for software process improvement is the relative lack of credible measurement data. If a company does not have the data to build the business case, it does not have the improvement project to get the data. It is the classic chicken-and-egg dilemma. But there is a solution.

PROBLEM DEFINITION

A 100-person software development team is responsible for a major software product containing a total of about 4,000,000 statements. The product has been built in a series of releases, with each release typically adding 800,000 to 1,200,000 statements. The development cycle for each release is approximately one year (which includes design, coding and all testing prior to release). The development team is responsible for all support and defect repair during the first year after release.

Hence, the team is concurrently responsible for maintenance of the previous release and development of the next. After the first year, support and defect repair is handled by a separate maintenance organization.

What needs to be known – "as-is" (baseline) and "to-be:"

(iv) The number of defects found in each phase – code, integration test, and system/acceptance test and release Year 1).
(v) The effort required to find and fix a defect in each phase.
(vi) Average labor rate for the team.

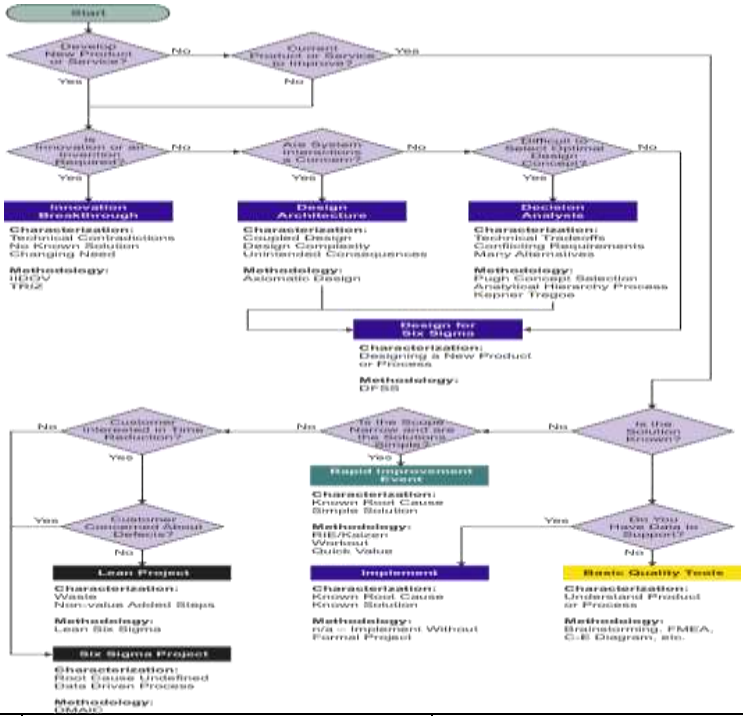
What is known?

1 The number of customer calls related to the prior release.
1 The number of internally reported defects.

- The approximate start and end date of each phase (although there was some overlap), and the percent of total effort devoted to maintenance of the prior release and development of the current release during those time periods.

Flow chart for the Problem Solving Methodology

The above leads to the following initial business case as outlined in Table 1.



	Rates	As-Is Baseline			To-Be			
Labor Rate (Per Hour)	**80.00**	**Defects**	**F&F Hours**	**Rupees**	**Defects**	**F&F Hours**	**Rupees**	**F&F Hours**
Total	**% Found**	16,510			16,510			

Code Inspections	18.0%	0			2,972	11,887	950,976	4
Integration Test	25.4%	4,188	70,615	5,649,224	3,434	41,213	3,297,056	12
System/Acceptance Test	48.1%	5,927	106,680	8,534,400	4,860	87,478	6,998,208	18
Customer (One Year)	38.7%	6,395	268,590	21,487,200	5,244	220,244	11,639,885	42
	Year 1 TCE	**61.3%**			**Year 1 TCE**	**68.2%**		
	Totals		**445,885**		**35,670824**	**360,822**	**28,865,744**	
						Savings	**6,805,080**	**19.1%**

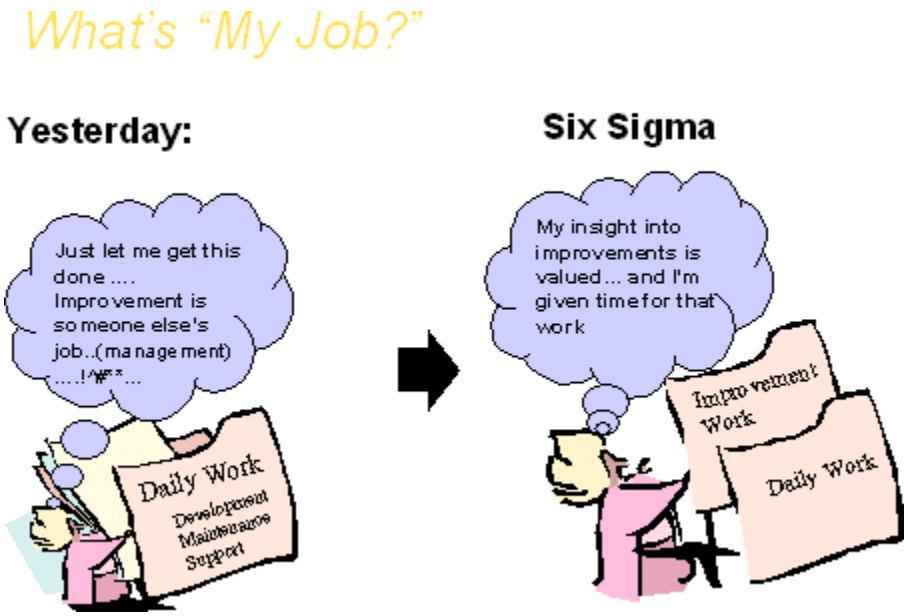
The pilot results were actually somewhat **better than the initial business case**. The percent of defects discovered by inspections was actually higher than forecast (25 percent versus 18 percent). This is based on the assumption that the total number of statements and the number of defects ‑inserted‖ remains the same compared to the previous release. Since no other process changes were made during this time and the staff is the same, this was a reasonable assumption – it will be check against actual results at the end of Year 2. The scenario presented in Table 2 assumes that the primary goal is to reduce cost. The business case in Table 3 assumes the hours devoted to testing will be unchanged compared to the baseline, but testing will be 10 percent less effective – i.e., the defects found in each test phase will be 10 percent less than in the baseline. Savings are only slightly less, but delivered quality is much higher as measured by TCE – 80.1 percent rather than 70.9 percent. About half as many defects are delivered – 1,967 rather than 3,837.

	Rates	**As-Is Baseline**			**To-Be**			
Labor Rate (Per Hour)	**80.00**	**Defects**	**F&F Hours**	**Rupees**	**Defects**	**F&F Hours**	**Rupees**	**F&F Hours**
Total	**% Found**	9,906			9,906			
Code Inspections	18.0%	0			2,477	9,906	792,480	4
Integration Test	25.4%	2,513	37,343	2,987,454	2,262	37,343	2,987,454	10
System/Acceptance Test	48.1%	3,556	90,678	7,254,240	3,200	90,678	7,254,240	26
Customer (One Year)	38.7%	3,837	161,154	12,892,320	1967	82,631	4,367,038	42
	Year 1 TCE	**61.3%**			**Year 1 TCE**	**80.1%**		
	Totals		**289,175**		**23,134,014**	**220,558**	**17,644,638**	
						Savings	**5,489,376**	**23.7%**

Based on the results so far, management has agreed to apply inspections to the complete product during the next release development cycle. They have also agreed to have the complete team use the new data collection tools and processes so that in the future accurate data will be available for the entire life cycle, including customer use. That data can be used to prepare much more accurate business cases for future improvement proposals, such as improvements to the test processes.

6. IMPROVEMENT OF PROCESS IN SIX SIGMA

Improving a process, like building character, can be done by the people involved, but not to them. Hence in Six Sigma we engage and empower the people who perform the software processes to plan and implement improvements themselves, with the guidance and assistance of Six Sigma specialists who are fully versed in software development best practices.
The Six Sigma DMAIC approach to process improvement provides a powerful mechanism for improving the software process. Typically benefits will exceed cost within 6 to 12 months from initiation of a Six Sigma program for software development, and the on-going return will be very substantial – often a 15-25 percent reduction in software development costs in year two, with continuing reductions thereafter.



7. Conclusion

The benefits of Six Sigma are great. Companies reported increased profitability and employee and customer satisfaction. Based on the findings of this study, we can conclude that benefits such as trained quality professionals in statistical control, increased profitability, improved employee job satisfaction, and success in quality components are important reasons to deploy Six Sigma in companies.

REFERENCES

1) https://en.wikipedia.org/wiki/Six_Sigma
2) De Feo, Joseph A.; Barnard, William (2005). JURAN Institute's Six Sigma Breakthrough and Beyond – Quality Performance Breakthrough Methods. Tata McGraw-Hill Publishing Company Limited. ISBN 0-07-059881-9.
3) Webber, Larry; Wallace, Michael (15 December 2006). Quality Control for Dummies. For Dummies. pp. 42–43. ISBN 978-0-470-06909-7. Retrieved 2012-05-16.
4) The Certified Quality Engineer Handbook, Third Edition, ed. Connie M. Borror, ASQ Quality Press, 2009, pp. 321–332.
5) https://www.isixsigma.com/new-to-six-sigma/dmaic/six-sigma-software-development-case-study/Gary A. Gack
6) "The Inventors of Six Sigma". Archived from the original on 2005-11-06. Retrieved 2006-01-29.
7) Tennant, Geoff (2001). SIX SIGMA: SPC and TQM in Manufacturing and Services. Gower Publishing, Ltd. p. 6. ISBN 0-566-08374-4.
8) "The Evolution of Six Sigma". Retrieved 2012-03-19.
9) https://www.isixsigma.com/new-to.../building-business-case-software-defect-reduction.
10) https://www.isixsigma.com/

*****