A Framework for the Use of Six Sigma Tools in PSP/TSP

Youngkyu Park, Hojin Choi, Jongmoon Baik
Information and Communications University
119, Munji-ro, Yuseong-gu, Deajeon, 305-714, Korea
{youngkyupark,hjchoi,jbaik }@icu.ac.kr

Abstract

The advent of software process models such as CMM/CMMI (Capability Maturity Model/Capability Maturity Model Integration) has helped software engineers understand principles and approaches of software process improvement. There, however, has been difficulty increasing productivity from applying those models since "how" is not within the scope of the CMM/CMMI. For this reason, SEI (Software Engineering Institute) introduced PSP/TSP (Personal Software Process/Team Software Process); however, they still lack statistical analysis tools and systematic process control techniques for analyzing measures collected in PSP/TSP. Six Sigma, on the other hand, provides the quantitative analysis tools necessary to identify high leverage activities, control process performance and evaluate effectiveness of process changes. Deploying PSP/TSP in conjunction with Six Sigma, therefore, can directly lead to improved project performance and continuous process improvement by analyzing data, assessing process stability, and prioritizing improvements in PSP/TSP. Continuing with this rationale, a framework that guides how and where Six Sigma tools are considered in PSP/TSP is proposed.

1. Introduction

Software has long been one of the most difficult challenges faced by many businesses. The rate of failure has been high, a large amount of software resources are consumed by rework, and yet software is critical to success in our economy. The cost of hardware has decreased dramatically, and quality has increased by orders of magnitude. However, the cost and quality of software has not seen comparable improvements [1].

Many different responses to these problems have evolved in recent years such as PSP/TSP. PSP/TSP are process definitions that provide the necessary skills, discipline, and commitment required for successful software projects [2]. In PSP/TSP, software developers collect data and use it for process improvement. However, even though many data are collected, PSP/TSP are unable to analyze measures due to a lack of statistical analysis tools, unable to assess process stability due to a lack of systematic process control techniques, unable to focus on vital-few improvements due to a lack of tools for prioritizing improvements, and unable to make valid fact-based decisions due to a lack of tools for decision making. Consequently, effective and sustainable process improvement becomes difficult in PSP/TSP [3].

Six Sigma, on the other hand, is not a software development process definition - rather it is a far more generalized process for improving processes and products. The primary goal of Six Sigma is to increase customer satisfaction, and thereby profitability, by reducing and eliminating defects using statistical analysis and decision-making tools. Deploying PSP/TSP in conjunction with Six Sigma tools, therefore, can directly lead to improved project performance and effective process improvement by analyzing data, assessing process stability, and prioritizing improvements [1].

There have been some efforts to use Six Sigma tools in PSP/TSP. They mostly look into relationships between Six Sigma and PSP/TSP, and yet some provide a few examples on how to use Six Sigma tools in PSP/TSP [1, 3]. However, it has not been easy for software developers or teams to use Six Sigma tools in the right place at the right time in PSP/TSP based on previous works because each Six Sigma tool requires different data from different PSP/TSP activities. For this reason, a framework that guides where and how Six Sigma tools are used in PSP/TSP should be given to software developers so that effective and continuous process improvement can be achieved in PSP/TSP.

The remainder of this paper is organized as follows. Section 2 gives an introduction to PSP, TSP and Six Sigma. Section 3 describes a framework for applying Six Sigma tools to PSP/TSP. For the definition of the framework, we identified Six Sigma tools for PSP/TSP

0-7695-2867-8/07 $25.00 © 2007 IEEE
DOI 10.1109/SERA.2007.57

and characterized the selected Six Sigma tools by selecting a set of PSP/TSP activities. Finally, we present conclusion and future work in section 4.

2. Background

2.1. PSP

PSP provides software engineers with a disciplined personal framework for doing software work. PSP process consists of a set of methods, forms, and scripts that show software developers how to plan, measure, and manage their work. PSP is designed for use with any programming languages or design methods, and it can be used for most aspects of software work, including writing requirements, doing tests, defining processes, and fixing defects.

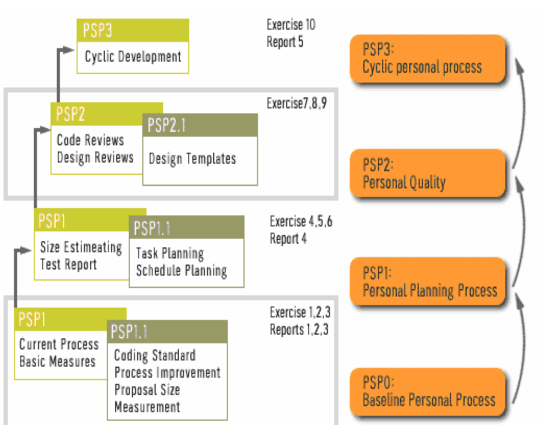

Figure 1. PSP Process Evolution

As shown in Figure 1, software engineers follow prescribed methods, represented as levels PSP0 through PSP3, and write a set of 10 programming exercises and five reports. With each exercise, they are gradually introduced to many advanced software engineering methods. By measuring their own performance, the engineers can see the effect of these methods on their work. When software engineers use PSP, recommended process goal is to produce products with no defect on schedule and within planned costs [4].

2.2 TSP

TSP was developed in the late 1990s to add team-level practices to PSP. By doing so, TSP makes PSP suitable for use in a commercial software development environment. TSP begins with a facilitated project launch process that generates a detailed project plan. The project plan includes a development strategy, a tailored development process, detailed size and effort estimates, earned value plans, a schedule, a quality management plan, and a risk management plan [5]. The launch process is team-building exercise designed to produce a high-performance work team. TSP continues to support project execution activities through a structured weekly project status meeting and all the management practices necessary to run a full-scale development project. The main benefit of TSP is that it shows software developers how to produce quality products for planned costs and on planned schedules. It does this by showing developers how to manage their work and by making them owners of their plan and process. TSP provides team projects with guidance on how to accomplish their objectives.

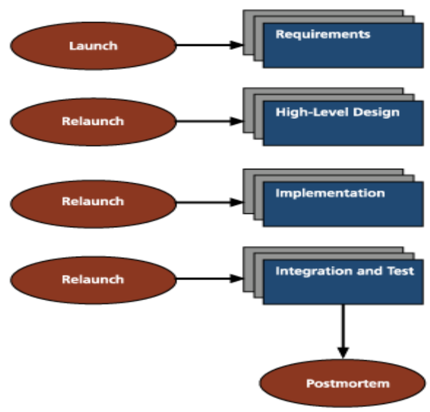

Figure 2. TSP Structure

As shown in Figure 2, TSP guides teams through the four typical phases of a project. These projects may start or end on any phase, or they can run from beginning to end. Before each phase, the team goes through a complete launch or relaunch, where they plan and organize their work. Typically, once team members are PSP trained, a four-day launch workshop provides enough guidance for the team to complete a full project phase. Then teams need a two-day relaunch workshop to start the second and each subsequent phase. These launches are not training, but they are part of the project.

2.3 Six Sigma

Six Sigma is a quality improvement approach to enhancing organization's performance by using statistical analytic techniques [6]. Six Sigma aims to eliminate the variability and defects which interfere with customer satisfaction and cost reduction. Six Sigma has been being embodied in the management strategy for quality improvement to quantitatively evaluate organization's processes and to reduce process variability [7].

Six Sigma is defined at three levels:

- Philosophy: Being more profitable, Six Sigma can be used for improving customer satisfaction by reducing and eliminating defects.
- Metrics: As a metric, Six Sigma equals to 3.4 defects per million opportunities (DPMO). Additionally Six Sigma includes several metrics such as Defect rate (parts per million), Sigma Level, Defects per Unit (DPU), and Yield.
- Improvement Framework: Six Sigma owns various toolkits and structured problem solving roadmaps such as DMAIC (Define, Measure, Analyze, Improve, Control) and DMADV (Define, Measure, Analyze, Design, Verify).

Six Sigma, in other words, is defined as the scientific quality improvement approach that measures the performance of the organization's processes and analyzes the cause of the defects by using Six Sigma roadmaps and toolkits, then eliminates the defects and pursues continuous, measurable, and controllable improvement of the organization's processes so as to accomplish Six Sigma level. By eliminating defects and process variability, Six Sigma achieves cost reduction and customer satisfaction.

3. A Framework for Applying Six Sigma Tools to PSP/TSP

3.1 A Set of Six Sigma Tools for PSP/TSP

It has been known that there are no standard Six Sigma toolsets in Six Sigma community [3, 8]. In fact, toolsets vary by approach, company, and organization. In addition, a few Six Sigma tools, such as DOE (Design of Experiment), MSE (Measurement System Evaluation), and lean manufacturing, have less applicability to PSP/TSP [9]. Therefore, it has been necessary to identify a set of selected Six Sigma tools for PSP/TSP for our purpose of applying Six Sigma tools to PSP/TSP.

For selecting Six Sigma tools for PSP/TSP, various elements in PSP/TSP process have been taken into consideration. The elements could be anything from measures, estimation models, and to the objectives of PSP/TSP if they need to be analyzed and enhanced by some means. 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 PSP/TSP. For instance, defects data collected throughout PSP/TSP process need to be prioritized by defects type 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 the future. For that purpose, Pareto analysis has the functionality for prioritization of defects types; therefore, it is included in a set of Six Sigma tools for PSP/TSP. Table 1 shows PSP/TSP elements that need to be analyzed for effective and continuous process improvement in the left-most column, how they should be analyzed in the middle, and finally Six Sigma tools that provide corresponding functionalities in the right-most column. As shown Table 1, 13 Six Sigma tools have been selected as Six Sigma tools for PSP/TSP. They all give functionalities for analyzing the elements of PSP/TSP. Therefore, they are useful for improving PSP/TSP process continuously and effectively.

It should be noted that some of the selected Six Sigma tools for PSP/TSP are team-based tools. Since PSP is a framework to help individual software developers improve their performance, team-based techniques are not applicable to PSP. The identified team-based tools are affinity diagram, SQFD (Software Quality Function Deployment), Kano analysis and SWFMEA (Software Failure Mode and Effect Analysis). Moreover, process mapping has less applicability to PSP because PSP has no complex process to be documented visually by process mapping. Therefore, Six Sigma tools for PSP are Pareto analysis, cause and effect diagram, control charts, ANOVA (Analysis of Variance), two-sample t-test, scatter plots, correlation analysis, and regression analysis. On the other hand, all of the selected Six Sigma tools for PSP/TSP except for ANOVA are applicable to TSP. ANOVA has less applicability to TSP because ANOVA can be used to determine goodness of "fit" of linear regression model, whereas linear regression model is not within the scope of TSP.

Table 1. PSP/TSP Elements and Corresponding Six Sigma Tools

PSP/TSP Elements		Functions	Corresponding Six Sigma Tools
PSP/TSP measures	Defect measure	Prioritization of defects type to find most frequent defects	Pareto analysis
		Identification of the cause of the most frequent defects	Cause and effect diagram

		Assessment of process stability	Control charts
	Other measures	Analysis of relationships between measures	Scatter plots, correlation analysis, regression analysis
PROBE		Determination of goodness of "fit" of PROBE	ANOVA
Process		Documentation of process flow to make complex process easier to understand	Process mapping
Customer needs		Requirements elicitation, prioritization, and analysis	Kano analysis, SQFD, affinity diagram
Estimation data		Determination of estimation accuracy	Two-sample t-test
Project risks		Risk assessment	SWFMEA

3.2 Correspondence between Six Sigma Tools and PSP/TSP Activities

The relationships between Six Sigma tools for PSP/TSP and PSP/TSP activities should be defined for the definition of the framework. Since Six Sigma tools for PSP/TSP are identified, PSP/TSP activities that are enhanced by the same Six Sigma tool should be grouped together. This work has been conducted based on input of each Six Sigma tool. In detail, input parameters of Six Sigma tools in terms of PSP/TSP process have been analyzed to identify types of PSP/TSP activities each Six Sigma tool corresponds to. Then we identified PSP/TSP activities necessary to apply identified Six Sigma tools for PSP/TSP. In other words, those identified PSP/TSP activities provide input necessary to apply Six Sigma tools, and thereby can be enhanced.

When grouping PSP/TSP activities, PSP/TSP phase and step names have been used. However, it should be noted that in TSP each phase has 2 scripts, one for cycle 1 and the other for cycle n. In addition, activities in cycle n are somewhat different from those in cycle 1; therefore, differences between two scripts should be taken into account when corresponding Six Sigma tools to TSP activities. For TSP, we pointed out specifically whether TSP activities enhanced by Six Sigma tools are from cycle 1 or n.

The following describes input of each Six Sigma tool for PSP/TSP and how it is used in PSP/TSP. PSP/TSP phases and steps each Six Sigma tool corresponds can be found in Figure 3 and Figure 4.

Cause and effect diagram

- Input: defects data, undesired effects
- Description: cause and effect diagram can be used to analyze defects data and identify causes of defects in Postmortem in PSP/TSP. Also, it is useful to identify possible causes of any problems with the prior cycle in PSP/TSP.

ANOVA

- Input: linear regression model
- Description: In PSP, the goodness of "fit" of PROBE can be determined by examining the mean-squared error in the ANOVA.

Scatter plots, correlation analysis, and regression analysis

- Input: two measures
- Description: these tools can be used to identify a correlation between measures collected in PSP/TSP. In PSP/TSP, these can be applied to identify a correlation between review rate and defects/KLOC. They can also be used to identify a relationship of A/FR to test defects.

Control charts

- Input: defects data
- Description: control charts are used to see whether the process is stable and consistent in PSP/TSP. For example, during Code Review, if defect density of some modules is beyond the UCL, they probably need to be rejected from reviews and should be reviewed again after rework.

Pareto analysis

- Input: defects types and their frequency
- Description: Pareto analysis can be used to devise defect prevention strategies in Postmortem in PSP/TSP. To come up with defect prevention strategies, defects types to focus on need to be selected with a Pareto chart of your defects.

Two sample t-test

- Input: size and time estimation data
- Description: two-sample t-test can be used to determine accuracy of the estimation made in PSP/TSP.

Kano analysis

- Input: product functions

- Description: Kano analysis can be used to prioritize product functions based on their impact to customer satisfaction in TSP.

Affinity diagram, SQFD
- Input: user requirements
- Description: affinity diagram can be used to elicit and analyze software requirements and to build a shared understanding of the user requirements in TSP. SQFD are useful for Requirements Tasks step in TSP. Once customer requirements are solicited and recorded, they are converted to technical and measurable statements of the software product. By following SQFD process, we can measure how important each statement is and find what design and integration risks and opportunities are.

SWFMEA
- Input: project risks
- Description: SWFMEA can be used to reduce project risks in TSP. SWFMEA should be a "living" document, updated whenever new errors are uncovered or when risk mitigation plans are modified.

Process mapping
- Input: process
- Description: when identifying configuration control board and its procedure, functional process map for current process can be developed to clearly delineates individual responsibilities and where hand-offs occur in TSP. Once the preliminary "as is" process map is created, the team should review it with all affected groups to ensure that the map accurately reflects the flow of responsibility.

The correspondence between Six Sigma tools for PSP/TSP and PSP/TSP activities is represented in Figure 3 and Figure 4. A set of selected Six Sigma tools for PSP/TSP are listed on the left-hand side and the PSP/TSP activities on the right-hand side. An arrow links each Six Sigma tool to the respective grouping of PSP/TSP activities. The right-hand side only shows PSP/TSP phases and steps enhanced by the application of Six Sigma tools.

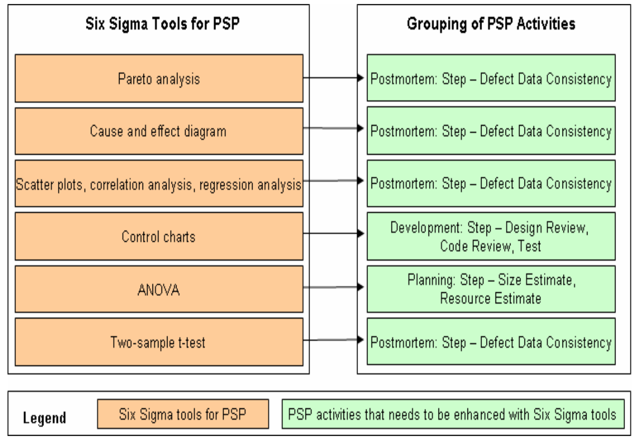

Figure 3. Correspondence between Six Sigma Tools for PSP and PSP Activities

As shown in Figure 3 and Figure 4, the greatest degree of correspondence between Six Sigma tools and PSP/TSP activities lies in Postmortem phase in which analysis of process data is conducted. This is mainly because purpose of many Six Sigma tools is to analyze data.

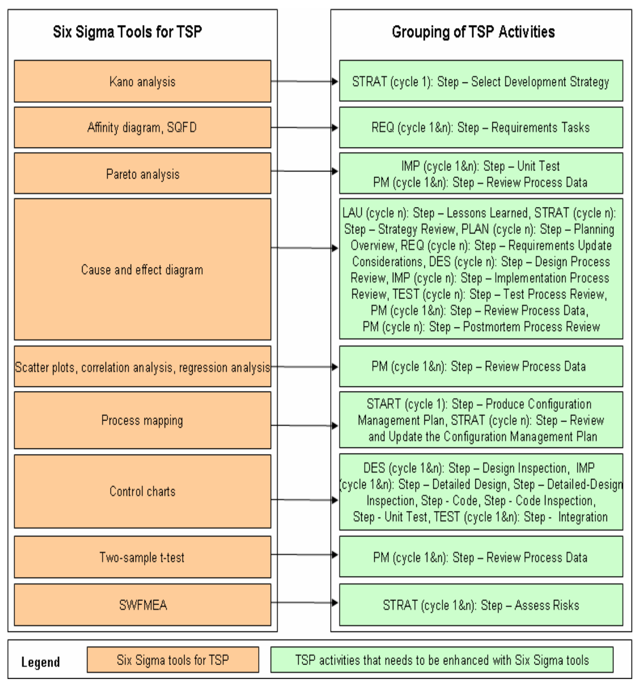

Figure 4. Correspondence between Six Sigma Tools for TSP and TSP Activities

3.3 A Framework: Two Views

A framework allows software developers or teams to take decisions about the specific Six Sigma tools to include into their PSP/TSP in order to achieve effective process management and improvement. It condenses all the information developers need to apply the solution proposed in this paper. In order to ease the usage of the framework by its potential users, we offer two views.

- View by Six Sigma tools: This view is organized by the name of the selected Six Sigma tools. It is useful when developers already know about a particular Six Sigma tool they have heard of. Accessing the framework through this view offers the possibility of knowing the characterization of a particular Six Sigma tool.
- View by PSP/TSP process: when developers try to find appropriate Six Sigma tools for activities in PSP/TSP, they access the framework through this view.

Table 2 and Table 3 show a framework for the use of Six Sigma tools in PSP/TSP with view by PSP/TSP process. This paper only presents a framework with view by process because two views basically represent the same information in different way.

Table 2. A Framework for the Use of Six Sigma Tools in PSP: View by PSP Process

PSP Phase	PSP Step	Six Sigma Tools	Purpose	Basic Reference
Planning	Size Estimate Resource Estimate	ANOVA	Determination of the goodness of "fit" of PROBE	[Pyzdek, 03]
Development	Design Review Code Review Test	Control charts	Statistical process control, evaluation of the significance of a process change	[WC, 92]
Postmortem	Defect Data Consistency	Cause and effect diagram	Identification of root causes	[Brassard, 98]
		Scatter plots	Identification of a correlation between two measures	[Pyzdek, 03]
		Two-sample t-test	Determination of estimation accuracy	[Miller, 86]
		Correlation analysis	Test of statistical significance of the relationship between two measures	[Pyzdek, 03]
		Regression analysis	Description of relationship between two measures precisely by means of an equation that has predictive value	[Pyzdek, 03]
		Pareto analysis	Prioritization of defects types	[Scholtes, 88]

Table 3. A Framework for the Use of Six Sigma Tools in TSP: View by TSP Process

TSP Phase	TSP Cycle	TSP Step	Six Sigma Tools	Purpose	Basic Reference
Team Launch	1	Lessons Learned	Cause and effect diagram	Identification of root causes	[Brassard, 98]
Development Strategy	1	Select Development Strategy	Kano analysis	Prioritization of product functions	[KNS, 84]
	1&n	Assess Risks	SWFMEA	Identification of possible risks	[RD, 79]
	n	Strategy Review	Cause and effect diagram	Identification of root causes	[Brassard, 98]
	1	Produce Configuration Management Plan	Process mapping	Documentation of process flow	[Damelio, 96]

	n	Review and Update the Configuration Management Plan			
Development Plan	n	Planning Overview	Cause and effect diagram	Identification of root causes	[Brassard, 98]
Requirements Development	1&n	Requirements Tasks	Affinity diagram	Requirements elicitation and analysis	[Beyer, 98]
	n	Requirements Update Considerations	Cause and effect diagram	Identification of root causes	[Brassard, 98]
	1&n	Requirements Tasks	SQFD	Requirements analysis	[HRS, 96]
Design	n	Design Process Review	Cause and effect diagram	Identification of root causes	[Brassard, 98]
	1&n	Design Inspection	Control charts	Statistical process control	[WC, 92]
Implementation	n	Implementation Process Review	Cause and effect diagram	Identification of root causes	[Brassard, 98]
	1&n	Unit Test	Pareto analysis	Prioritization of test cases	[Scholtes, 88]
	1&n	Detailed Design	Control charts	Statistical process control	[WC, 92]
	1&n	Detailed-Design Inspection			
	1&n	Code			
	1&n	Code Inspection			
Integration and System Test	1&n	Test Process Review	Cause and effect diagram	Identification of root causes	[Brassard, 98]
	1&n	Integration	Control charts	Statistical process control	[WC, 92]
Postmortem	n	Postmortem Process Review	Cause and effect diagram	Identification of root causes	[Brassard, 98]
	1&n	Review Process data	Scatter plots	Identification of a correlation between two measures	[Pyzdek, 03]
			Correlation analysis	Test of statistical significance of the relationship between two measures	[Pyzdek, 03]
			Regression analysis	Description of relationship between two measures precisely by means of an equation that has predictive value	[Pyzdek, 03]
			Two-sample t-test	Determination of estimation accuracy	[Miller, 86]
			Pareto analysis	Prioritization of defects	[Scholtes, 88]

4. Conclusion and Future Work

In this paper, we proposed a framework for the use of Six Sigma tools in PSP/TSP. For the definition of the framework, we have first selected a set of Six Sigma tools for PSP/TSP, and then characterized the Six Sigma tools by selecting a set of PSP/TSP activities so that software developers or teams know how and where in PSP/TSP Six Sigma tools are used. Introducing this framework to PSP/TSP, software development organizations can possibly improve customer satisfaction by reducing defects, and increase profits by eliminating variability and improving quality.

Performance increase at the individual and team level can be attained by identifying issues and analyzing measures that could happen at team level, and that provides a basis for applying Six Sigma tools at the project and organization level. For future work, we will extend the use of Six Sigma tools and methodologies to the organization and business level to create more values and better customer satisfaction. In addition, resources in software development projects are usually scarce. Therefore, Six Sigma tools need to be characterized in terms of their values to software organizations so that the software organizations can select a subset of selected Six Sigma tools for PSP/TSP when they have scarce resources. A tailoring guideline that helps software development organizations choose a subset of the Six Sigma tools for PSP/TSP based on ROIs of Six Sigma tools will be proposed in the future.

5. Acknowledgement

"This research was supported by the 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-2006-(C1090-0603-0032)).

6. References

[1] Gack A. and Robinson K., "Integrating Improvement Initiatives: Connecting Six Sigma for Software, CMMI, Personal Software Process and Team Software Process", Software Quality Journal 5 (4), 5-13, 2003.
[2] N. Davis and J. Mullaney, "The Team Software Process (TSP) in Practice: A Summary of Recent Results", Technical Report CMU/SEI-2003-TR-014, Software Engineering Institute, http://www.sei.cmu.edu/publications/, 2003.
[3] Janieszewski, S. and George, E., "Integrating PSP, TSP and Six Sigma," Software Quality Professional, Vol. 6, No. 4, pp.4–13, 2004.
[4] Watts S. Humphrey, "The Personal Software Process (PSP)", Software Engineering Institute, CMU/SEI-2000-TR-022, ESC-TR-2000-022, 2000.
[5] Watts S. Humphrey, "The Team Software Process (TSP)", Software Engineering Institute, CMU/SEI-2000-TR-023, ESC-TR-2000-023, 2000.
[6] Thomas Pyzdek, The Six Sigma Handbook: The Complete Guide for Greenbelts, Blackbelts, and Managers at All Levels, Revised and Expanded Edition, McGraw-Hill, 2003.
[7] Pete, Larry Holpp, What is Six Sigma?, McGraw-Hill, 2002.
[8] Dan S. Van Duine, Advanced Information Services, Inc.,"Experiences Integrating PSP and TSP with Six Sigma," Team Software Process Symposium, 2006.
[9] Mahanti, Rupa; Antony, Jiju, "Confluence of six sigma, simulation and software development," Managerial Auditing Journal, Volume 20, pp. 739-762(24) Number 7, 2005.
[Damelio, 96] Damelio, Robert, "The Basics of Process Mapping," Portland, Oregon: Productivity Inc, 1996.
[Scholtes, 88] Scholtes, P.R. et al., "The Team Handbook," p. 2-25. Madison, WI: Joiner Associates, 1988.
[Brassard, 98] Brassard, M., "The Memory Jogger: A Pocket Guide of Tools for Continuous Improvement", pp. 24 - 29. Methuen, MA: GOAL/QPC, 1988.
[Beyer, 98] Beyer, H., Holtzblatt, K., "Contextual Design: Defining Customer-Centered Systems", Morgan Kaufmann, 1998.
[HRS, 96] Haag, S., M.K. Raja and L.L. Schkade, " Quality Function Deployment Usage in Software Development," Communications of the ACM, 39(l), pp. 41-49, 1996.
[KNS, 84] Kano, Noriaki, Seraku, N., Takahashi, F., and Tsuji, S., "Attractive and Normal Quality," Quality, (14:2), pp. 39-48, 1984.
[RD, 79] Reifer, Donald J., "Software Failure Mode and Effects Analysis," IEEE Transactions on Reliability, vol.28, no.3, August 1979.
[WC, 92] Wheeler, D.J., & Chambers, D.S., "Understanding Statistical Process Control (2nd Ed.)", Knoxville, TN: SPC Press, 1992.
[Pyzdek, 03] Thomas Pyzdek, "The Six Sigma Hand book (II Ed.)", McGraw Hill, 2003.
[Miller, 86] Miller, Rupert G. Jr., "Beyond ANOVA: Basics of Applied Statistics", New York: John Wiley & Sons, 1986.