A Six Sigma Framework for Software Process Improvements and its Implementation

Zhedan Pan, Hyuncheol Park, Jongmoon Baik, Hojin Choi
School of Engineering, Information and Communications University,
119 Munji-ro, Yuseong-Gu, Daejeon, Republic of Korea

{panzhedan, hcparker, jbaik, hjchoi}@icu.ac.kr

Abstract

Six Sigma has been adopted by many software development organizations to identify problems in software projects and processes, find optimal solutions for the identified problems, and quantitatively improve the development processes so as to achieve organizations' business goals. A Six Sigma framework for software process improvements is needed to provide a standard process and analysis tools for Six Sigma project executions, and also provide a platform for collaborations with other process improvement approaches, such as PSP/TSP and CMM/CMMI. However, few frameworks have been proposed to support Six Sigma project executions. Most of Six Sigma projects for software process improvements have been performed in an ad-hoc way. In this paper, we propose a framework to support Six Sigma projects for continuous process improvements for software developments. Based on this framework, we implemented a web-based tool, called SSPMT integrated with a software project management tool and a PSP supporting tool. The suggested framework and SSPMT is beneficial in initiating and executing Six Sigma projects, facilitating data collection and data analyses by Six Sigma toolkits, and standardizing the Six Sigma project execution process so as to achieve Six Sigma project goals and of organizations' business goals.

1. Introduction

Motivated by the theme that product quality is determined by the process which produces the product [8], 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. Software Six Sigma was introduced to focus on the adoption of Six Sigma methodologies to improve software development processes. Software Six Sigma concerns all aspects of software development life cycle and targets low cost improvements to the organizations with immediate returns so that highest benefits can be achieved in short period of time. It also requires the integration with other software process improvement approaches such as PSP (Personal Software Process), TSP (Team Software Process), and CMM/CMMI (Capability Maturity Model/Capability Maturity Model Integration) [8,14,15,16].

However, currently there are few Six Sigma frameworks available to facilitate the Six Sigma project execution. Most of Six Sigma projects for software process improvements have been performed in an ad-hoc way.

In this paper, we provide a conceptual model of Software Six Sigma. Based on this, we propose a framework to support Six Sigma projects for software process improvements. This framework is built on Six Sigma DMAIC (Define, Measure, Analyze, Improve, and Control) methodology, Six Sigma analysis toolkits, data repository, and integration with other applications such as Jasmine [5] and SymphonyPM [6]. We implement this framework by developing a web-based tool called SSPMT (Six Sigma Project Management Tool). Steps to execute a Six Sigma project by SSPMT are described, and differences between Six Sigma projects and typical software projects are provided. Then we provide the recommended integration of Project data and PSP data with the analysis toolkits in

1530-1362/07 $25.00 © 2007 IEEE
DOI 10.1109/ASPEC.2007.43

the SSPMT which can provide easier execution of a Six Sigma project and data collection for various analyses.

This paper is organized as follows. In Section 2, background information with respect to Six Sigma and its DMAIC methodology are provided. In Section 3, The conceptual model of Software Six Sigma is introduced. Section 4 describes the Six Sigma framework for software process improvements and SSPMT. Section 5 provides execution steps of a Six Sigma project by SSPMT and differences between Six Sigma projects with typical software projects. In section 6, the recommended mapping of tools in SSPMT with Project data and PSP data is illustrated. Finally, the conclusion and future work are given.

2. Background

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

2.1. 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 [4]. 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 [9].

Six Sigma is described in terms of three perspectives [10]:

- 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 [11].
- Improvement framework: Six Sigma owns various toolkits and structured problem solving methodologies such as DMAIC and DFSS (Design For Six Sigma)

2.2. 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 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.

3. Conceptual Model for Software Six Sigma

Based on Six Sigma, Software Six Sigma [13] is introduced for the adoption of Six Sigma methodologies and toolkits in the software area, especially in reducing the defects in software development life cycles to improve the software product quality and increase customer satisfaction. Software Six Sigma covers all aspects of the software development life cycle with DFSS and DMAIC methodologies. With the help of various toolkits, Software Six Sigma can target low cost improvements to the organizations with immediate returns so that highest benefits can be achieved in a short period of time. It collaborates with other software process improvement approaches such as PSP, TSP, and CMM/CMMI [8,14,15,16]. The conceptual model of Software Six Sigma is illustrated in Figure 1.

In Figure 1, Software Six Sigma is inherited from Six Sigma and supported by PSP, TSP, CMMI and other approaches. As described in Section 2, Six Sigma has an improvement framework, tools, metrics and sigma levels, which are inherited by Software Six Sigma for process and product quality improvements. PSP includes phases and steps to support personal software process improvement. In each step, PSP data about quality and performance can be analyzed by Six Sigma tools. TSP includes phases, cycles and steps to support team software process improvement. In each step, TSP data can be analyzed by Six Sigma tools. CMMI includes levels, process areas and representations. Each process area can be supported by Six Sigma improvement framework and related data can be analyzed by Six Sigma tools. Other approaches can include Project Management, Analysis Method, SPICE (Software Process Improvement and Capability dEtermination), Software Life Cycle Process (IEEE 12207) and so on.

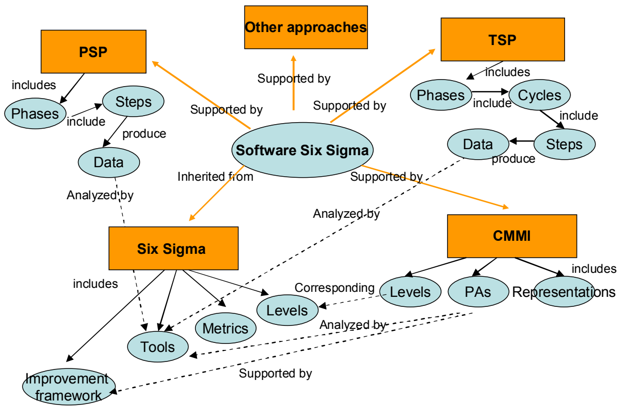


Figure 1. A Conceptual model for Software Six Sigma

This conceptual model for Software Six Sigma can be applied to execute Six Sigma projects for software process improvements. Aligning with business goals, a Six Sigma project for a software process improvement normally starts from a problem identified in the software development project, and then collects data, such as Project data and PSP data, to identify the root causes of the problem, and implements an optimal solution strategy to solve the problem.

4. Six Sigma Framework and SSPMT

In this section, we describe a Six Sigma framework for software process improvements, which can provide various capabilities to manage, support, control, track, and report the execution of Six Sigma projects for the improvements of the existing processes. It also provides a guideline about how to proceed a Six Sigma project as well as the relationships between the Six Sigma process flow and other applicable Six Sigma analysis tools and other applications.

In this section, we propose the framework in three points of view: architecture, integration with other applications, and SSPMT build on the framework.

4.1. Architecture

In order to support, manage and control Six Simga project executions, Six Sigma framework provides essentially required components, relationships of the components within the framework, and external components required to collect necessary data and analyze collected data effectively for process improvements. Figure 2 illustrates the architecture of the Six Sigma framework.

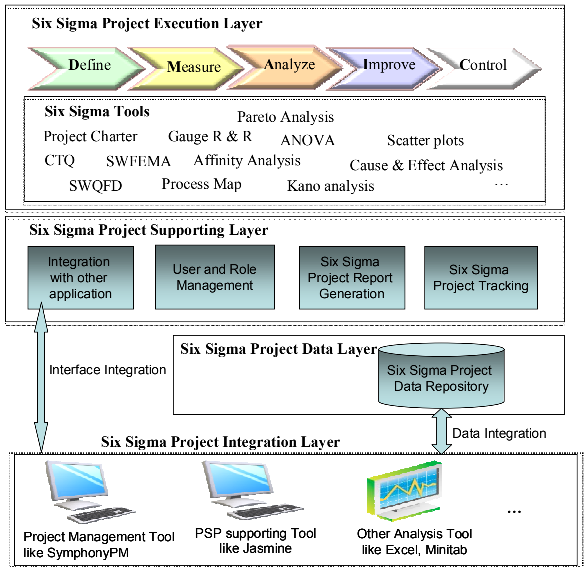

Six Sigma Project Execution Layer
Define
Measure
Analyze
Improve
Control
Six Sigma Tools
Pareto Analysis
Project Charter
Gauge R & R
ANOVA
Scatter plots
CTQ
SWFEMA
Affinity Analysis
Cause & Effect Analysis
SWQFD
Process Map
Kano analysis
…
Six Sigma Project Supporting Layer
Integration with other application
User and Role Management
Six Sigma Project Report Generation
Six Sigma Project Tracking
Six Sigma Project Data Layer
Six Sigma Project Data Repository
Interface Integration
Data Integration
Six Sigma Project Integration Layer
Project Management Tool like SymphonyPM
PSP supporting Tool like Jasmine
Other Analysis Tool like Excel, Minitab
…

Figure 2. Architecture of Six Sigma Framework

This Six Sigma framework consists of four layers: Six Sigma Project Execution Layer, Supporting Layer, Data Layer and Integration Layer. The components in these layers are as follows:

Six Sigma Project Execution Layer

- Six Sigma Project Execution Flow
  This component provides the DMAIC-based Six Sigma projects execution flow.
- Six Sigma Toolkits
  This component provides all the Six Sigma tools. More explanations are provided in later part of this Section.

Six Sigma Project Supporting Layer

- User and Role Management
  This component provides the management of users in the framework. Each user has his own

role like champion, master black belt, black belt or green belt.

- Six Sigma Project Tracking
  This component provides a traceability to track the status of ongoing Six Sigma projects based on organizations or projects.
- Six Sigma Project Report Generation
  This component creates a report associated with the analyzed results for the ongoing projects and completed projects.

Six Sigma Project Data Layer

- Six Sigma Project Data Repository
  This component provides a repository to store the collected data and the analyzed results during Six Sigma projects.

Six Sigma Project Integration Layer

- Integrations With Other Applications
  This component provides integration with other applications, such as PSP supporting tool and Project Management tool, which can complement Six Sigma framework by their own dedicated functions and data collection. More explanations are provided in Section 4.2.

Table 1. Selected toolkits in Six Sigma framework

Phase	Tools
Define	Project Charter Project Evaluation CTQ characteristics SWFMEA SWQFD Kano analysis Process map Affinity Diagram
Measure	Gage R&R Process capability analysis Process yield analysis
Analyze	Pareto analysis Cause and effect Diagram Correlation analysis Regression analysis Two sample t-test ANOVA Earned value analysis Scatter plots Line chart
Improve	Cause and Effect Diagram SWFMEA Cost Benefit analysis
Control	SWFMEA (Control) Control Chart Project Assessment Project Summary

DMAIC methodology within a Six Sigma project is the core part of the framework. A Six Sigma project includes several Six Sigma toolkits. These Six Sigma toolkits are highly required, effective, and efficient in conducting a Six Sigma project based on DMAIC methodology. Some researchers have tried to identify useful Six Sigma toolkits for software process improvements [1, 2, 7]. Table 1 illustrates the list of these Six Sigma toolkits which have been widely used at each phase of the DMAIC methodology. The Six Sigma framework includes those toolkits to support and manage Six Sigma projects.

4.2. Integrations with Other Applications

The Six Sigma framework can extend its functionalities through the integrations with other applications such as PSP supporting tool, project management tool, and analysis tools. The integration with other applications provides the flexibility to extend Six Sigma framework with other functionalities from those tools. The integration also helps Six Sigma to overcome its drawback: a lack of way to collect data from the current process. A PSP supporting tool and a project management tool can provide the sources of data which are required during Six Sigma projects. The description of the PSP supporting tool and the project management tool adopted are presented below.

- A PSP supporting tool: Jasmine [5]

Jasmine was developed to provide capabilities to collect reliable data automatically so as to support personal process improvement and quality management. Jasmine is mainly composed two subsystems: PPMT (Personal Process Management Tool) and PSPG/ER (PSP Guide/Experience Repository). PPMT client consists of sensors which can collect PSP data automatically. PSP data like defects, size and efforts can be collected automatically. It also supports the activities such as planning, earned value tracking, simple data analysis, and reporting. PSPG/ER provides the guidance of PSP process (such as PSP0, PSP0.1), activities (such as planning, design and coding) and artifacts (such as task plan and schedule).

- A project management Tool : SymphonyPM [6]

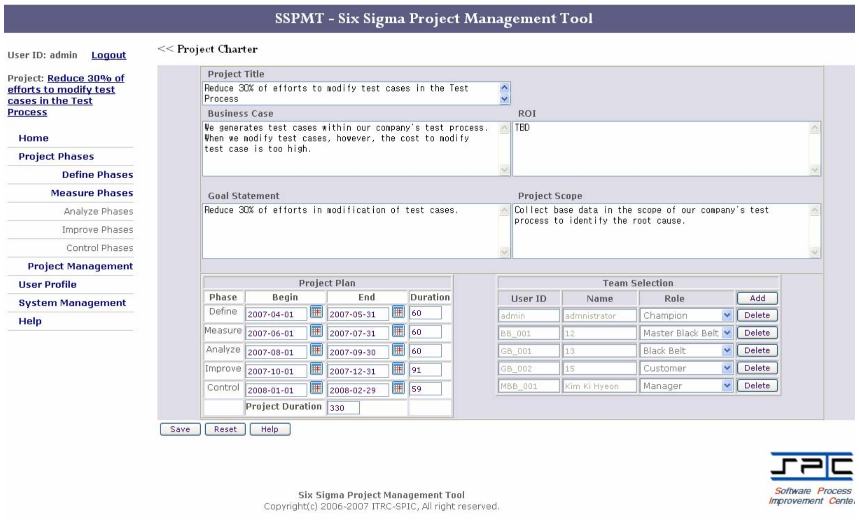
SymphonyPM is a CMMI_based Project Management Tool to support a Project Manager's main jobs – establishing, launching, monitoring, and evaluating project plan. The use of SymphonyPM helps project managers and members with the efficient software development project execution by providing a tailoring of a standard process and project resource assignment based on the tailored standard process.

4.3. Implementation

We have implemented the proposed Six Sigma framework as a web-based software tool, called SSPMT which supports and manages Six Sigma projects quantitatively and effectively. SSPMT can help organizations or Six Sigma project team members to manage, track, control, and report Six Sigma projects for software process improvements. SSPMT can lead Six Sigma projects through Six Sigma DMAIC methodology.

At the each phase of DMAIC, various Six Sigma toolkits are provided to conduct measurements and analyses for Six Sigma projects. These Six Sigma toolkits cover many aspects of a Six Sigma project, such as creating a project charter to initiate a Six Sigma project, establishing measurements to collect data and validate data by Gage R&R, analyzing the collected data by correlation analysis and regression analysis, identifying problems in the current process by using cause and effect diagram, improving the current process with risk analysis by SWFMEA (SW Failure Mode & Effects Analysis), and maintaining the improved process by control chart. Each tool has online helps to provide users with more detailed information about the usage of each tool.

Figure 3 shows a screenshot of SSPMT. The left side of the screenshot shows the DMAIC phase, and the right side shows Project Charter tool in Define phase. Project charter can be used for every Six Sigma project to define the project goals and scope, schedule, and project team members.


SSPMT - Six Sigma Project Management Tool

User ID: admin Logout

Project: Reduce 30% of efforts to modify test cases in the Test Process

Home
Project Phases
Define Phases
Measure Phases
Analyze Phases
Improve Phases
Control Phases
Project Management
User Profile
System Management
Help

<< Project Charter

Project Title
Reduce 30% of efforts to modify test cases in the Test Process

Business Case
We generates test cases within our company's test process. When we modify test cases, however, the cost to modify test case is too high.

ROI
TBD

Goal Statement
Reduce 30% of efforts in modification of test cases.

Project Scope
Collect base data in the scope of our company's test process to identify the root cause.

Project Plan			
Phase	Begin	End	Duration
Define	2007-04-01	2007-05-31	60
Measure	2007-06-01	2007-07-31	60
Analyze	2007-08-01	2007-09-30	60
Improve	2007-10-01	2007-12-31	91
Control	2008-01-01	2008-02-29	59
	Project Duration	330	

Team Selection			
User ID	Name	Role	Add
admin	admnistrator	Champion	Delete
BB_001	12	Master Black Belt	Delete
GB_001	13	Black Belt	Delete
GB_002	15	Customer	Delete
MBB_001	Kim Ki Hyeon	Manager	Delete

Save Reset Help

Six Sigma Project Management Tool
Copyright(c) 2006-2007 ITRC-SPIC, All right reserved.

Software Process Improvement Cente

Figure 3. Screenshots of SSPMT

5. Six Sigma Project Execution by SSPMT

In this section, firstly we provide a step-by-step execution of a Six Sigma project by using SSPMT. Then we summarize the differences between Six Sigma projects and typical software projects.

5.1 Steps to Execute a Six Sigma Project

SSPMT can facilitate the execution of a Six Sigma project for the software process improvement. The following provides a step-by-step description of executing a Six Sigma Project by SSPMT:

Define phase

1) Create a Six Sigma project: Six Sigma Champion can start a Six Sigma project by creating a Project Charter with information about project goals, scopes and project members.

2) Evaluate projects: There may have many Six Sigma projects needed to be executed, but only some of them can be selected and continued. Project Evaluation Tool is used to evaluate each Six Sigma project. Projects with the high scores can go into the next steps.

3) Listen Voice of Customers: Any Six Sigma project needs to listen to customers' voices to identify the most important factors for customers to achieve the big "Y". SWQFD (Software Quality Functionality Deployment), Kano analysis and affinity diagram can be used to identify, group and analyze the voice of the customers.

4) Build process map: Most Six Sigma projects need to define the current process of the targeted scope. Process Map Tool can be used to build current process.

5) Define failures modes: After building the process map, in each process, there may have some unsatisfied problems. SWFMEA can be used to define potential failure processes and failure modes for improvement.

6) Define CTQ: CTQ (Critical To Quality) refers to items which have most important impacts on the usage of software products. Defects and opportunities are needed to be defined for each CTQ item.

Measure:

7) Collect data: All the data related to current process and CTQ should be collected. Data collection can be manually and automatically. Since SSPMT provides the integrations with other applications like SymponyPM and Jasmine. Data in those tools can be collected automatically to SSPMT. Please refer to Section 6 for detail.

8) Validate data: All the collected data can be validated by Gage R&R.

9) Calculate process capability and yield: According to CTQ characteristics, and the collected data about defects and opportunities, each process capability can be calculated, and the whole process yield can also be calculated by multiplying each process capability.

Analyze:

10) Analyze statistical data: Statistical analysis tools such as Pareto analysis, Correlation analysis, Regression analysis, Two sample t-test, ANOVA, Earned value analysis, Scatter plots, and Line chart can be used to analyse the statistical data collected.

11) Analyze descriptive data: Cause and Effect Diagram Tool can be used to analyse descriptive data and find causes of the problems [12].

12) Identify root causes: Root causes of the problems needs to be identified with the help of analysis results from statistical analysis and Cause and Effect Diagram Tool.

Improve

13) Find solutions: Solutions to the identified root causes needs to be figured out. Cause and effect Diagram can be used to identify the potential solutions and their effects so as to ascertain the right solutions.

14) Select optimal solution: There may have many solutions to the problems. Cost and benefit analysis Tool can help to select the optimal solution according to ROI.

Control

15) Control the improvements: After the adoption of the solution, control chart can be used to control the improved process.

16) Assess the results: Assessment is to identify whether the Six Sigma project goal is achieved or not. All the tools in Measure and Analyze phase can be used to evaluate the results. SWFMEA (control) can also be used to evaluate the results. All the evaluated results can be collected into Project Assessment Tool.

17) Summarize lessons learned: All the lessons learned and best practices can be summarized into the Project Summary Tool.

18) Identify another Six Sigma project: After assessment and summary, it is possible to find some other improvements needed. Another Six Sigma project can be identified and be started for continuous improvements of processes and products.

These are reference steps used to explain how Six Sigma projects are executed in SSPMT. In the real practices, Six Sigma projects should be executed and tools should be selected according to each project's characteristics.

5.2 Differences between Six Sigma Projects and Typical Software Projects

From the above section, we can see that Six Sigma projects are different from typical software projects. The main five aspects of differences are described in the following:

- Start: Six Sigma projects start from an organization executives or Champions, while typical software projects start from customer requirements or marketing requirements.
- Team: Six Sigma projects are executed by Six Sigma project team, which may include champion, black belts and green belts, while typical software projects are executed by project manager and team members, which may include analysts, designers, developers and testers.
- Objectives: The objective of Six Sigma projects is to improve the current processes or products of organizations to achieve their business goals, while the objective of typical software projects is to produce software products to achieve organizations' business goals.
- Processes: Six Sigma projects adopt DMAIC or DFSS methodology to perform, while typical software projects have many software development processes to adopt such as Waterfall, Incremental process or RUP (Rational Unified Process).
- Toolkits: Six Sigma projects use Six Sigma toolkits (Table 1) to help identify and analyze problems of the products and process for improvements, while typical software projects use tools such as UML and Programming languages to produce software products.

6. Integration with Project Data and PSP Data

From Table 1, we can notice that there are not many Six Sigma tool to support data collections, and data needs to be collected manually. By adopting the proposed framework and SSPMT, data can be collected automatically from SymphonyPM and Jasmine.

For a typical software project, SymphonyPM records Project data of a project, such as the project time and processes, while Jasmine collects PSP data of each team member of the project, such as defects and efforts. These data are very valuable for executing a Six Sigma project and analyzing problems for the Six Sigma project. The mapping of these data to tools in SSPMT can make an easier conducting of Six Sigma projects.

In this section, we focus on how the Project data and PSP data are integrated and mapped into SSMPT.

6.1 Mapping Project Data with SSPMT

SymphonyPM can collect mainly three kinds of data: project overall information, process data, and activities data. *Project overall information* includes project scope, customers, objectivities, schedule, and development environment. *Process data* includes the start time, end time, and status of each process from the requirements, design, implementation and testing. *Activity data* includes tasks, schedule and resources to finish them.

Table 2. Recommended mapping of project data with tools in SSPMT

Project Data in SymphonyPM	**Tool in SSPMT**	**Description**
Project information	Project charter	To represent project information from SymphonyPM to project charter.
Process data	Process map	To represent processes and their statuses by process map.
Planned and actual time by process	Pareto analysis	To analyze the percentage of time spent by process. [7]
Activity data with time	Line chart	To analyze the efforts spent by each activity.
Time spent by each team member	Line chart	To analyze the efforts spent by each team member. [6]
Planned and actual time by activity	Two-sample T-test	To evaluate the estimation accuracy of two samples.

Table 2 shows the recommended mapping of Project data with tools in SSPMT. Project data can be mapped to tools for various analyses. Same data can be mapped to different tools for different purposes.

6.2 Mapping PSP Data with SSPMT

Jasmine can collect many PSP data, such as defects data in units test, compile and runtime, planned size and actual size in terms of SLOC (Source Lines Of Code), planned value, earned value, actual cost, planned finish time, and actual finish time. All these data can be mapped to SSMPT tools for analyses. Table 3 shows the recommended mapping of PSP data with tools in SSPMT to perform various analyses.

Table 3. Recommended mapping of PSP data with tools in SSPMT

PSP Data in Jasmine	**Tool in SSPMT**	**Description**
Defects	Trend analysis	To present defects distribution by the time.
	Scatter plots	To analyze the percentage of defects by defect type.[7]
Defects of all types	Pareto analysis	To analyze which defect type generates more defects. [5]
Planned and to-date SLOC	Two-sample T-test	To evaluate the estimation accuracy of two samples. [7]
Size and defects	Correlation analysis	To estimate size or defects of the programs.
	Regression analysis	To analyze the relationship between size and defects
Size and development time	Correlation analysis	To estimate size or development time.
	Regression analysis	To analyze the relationship between size and time.
Planned value, earned value, actual cost	Earned value analysis	To analyze whether current status of project is behind schedule or over budget.
Actual finish time and size	Correlation analysis	To analyze the relationship between time and size.

7. Conclusion and Future Work

Software Six Sigma has been adopted to help software organizations to improve their software processes so as to increase the software product quality. A Six Sigma framework for software process improvements can support and facilitate the adoption of Software Six Sigma. With this motivation, we developed the conceptual model of Software Six Sigma, and based on this conceptual model, we proposed the Six Sigma framework, which consists of Six Sigma

DMAIC methodology, Six Sigma toolkits, data repository, and integrations with other applications such as Jasmine and SymphonyPM. We implemented this framework with a web-based tool, called SSPMT. We also provided the steps to execute a Six Sigma project for software process improvements by using SSPMT and summarized the differences between Six Sigma projects and typical software projects. Then we analyzed the mapping of Project data and PSP data with tools in SSPMT. The suggested framework and SSPMT is beneficial in initiating and executing Six Sigma projects, facilitating data collection and data analyses by Six Sigma toolkits, and standardizing the Six Sigma project execution process so as to achieve Six Sigma project goals and of organizations' business goals.

SSPMT is still under development and testing. For future work, we will complete the development of the SSPMT and apply it to the real Six Sigma projects to prove its effectiveness and efficiency. Later, more software process improvement approaches, such as TSP and CMMI will be integrated with the proposed framework. To support a design of new process and products, DFSS will also be included in the framework in the future.

Acknowledgement

This research is 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))

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 Project Planner : A Step-by-Step Guide to Leading a Six Sigma Project Through DMAIC", McGraw-Hill, 2003.

[3] Rowland Hayler, "What is Six Sigma Process Management?", McGraw-Hill, 2005.

[4] Thomas Pyzdek, "The Six Sigma Handbook : A Complete Guide for Green Belts, Black Belts, and Managers at All Levels", McGraw-Hill, 2003.

[5] Hyunil Shin, Ho-Jin Choi, and Jongmoon Baik, "Jasmine: A PSP Supporting Tool", ICSP 2007, LNCS 4470, pp. 73–83, 2007.

[6] SymphonyPM, http://spic.kaist.ac.kr/leadspi.html

[7] Youngkyu Park, Hyuncheol Park, Hojin Choi, Jongmoon Baik, "A Study on the Application of Six Sigma Tools to PSP/TSP for Process Improvement", ICIS-COMSAR'06, 2006

[8] W. S. Humphrey, Managing the Software Process, Addison- Wesley, Reading, MA, 1989

[9] Pete, Larry Holpp, What is Six Sigma?, McGraw-Hill, 2002

[10] Jeannine Siviy, M. Lynn Penn, and Erin Harper, "Relationships Between CMMI® and Six Sigma", Software Engineering Institute, CMU/SEI-2005-TN-005, 2005

[11] David L. Hallowell, "Six Sigma Software Metrics, Part 3",http://software.isixsigma.com/library/content/c031015a.asp

[12] Zhedan Pan, Hoyeon Ryu, and Jongmoon Baik, "A Case Study: CRM Adoption Success Factor Analysis and Six Sigma DMAIC Application", *International Conference on Software Engineering Research, Management and Applications (SERA)*, 2007

[13] www.softwaresixsigma.com

[14] Watts S. Humphrey, "The Personal Software Process (PSP)", Software Engineering Institute, CMU/SEI-2000-TR-022, ESC-TR-2000-022, 2000

[15] Watts S. Humphrey, "The Team Software Process (TSP)", Software Engineering Institute, CMU/SEI-2000-TR-023, ESC-TR-2000-023, 2000

[16] CMMI, "Capability Maturity Model Integration", Software Engineering Institute, Carnegie Mellon University, 2002.