

# Quantitative Project Management Framework via Integrating Six Sigma and PSP/TSP

Sejun Kim, BISTel

Okjoo Choi, KAIST

Jongmoon Baik, KAIST

**Abstract:** Process technologies such as Personal Software Process<sup>SM</sup> (PSP) and Team Software Process<sup>SM</sup> (TSP) provide a good foundation for Six Sigma applications in business. Business approaches using Six Sigma provide methods for process improvement and analysis to achieve the goals of the PSP/TSP. This article discusses a framework with which software engineers and project managers can quantitatively manage software projects for improving the processes by applying Six Sigma in conjunction with PSP/TSP.

## Introduction

The advent of CMMI<sup>®</sup> has helped software engineers and project managers understand the principles and approaches of software process improvement [1, 2]. Many people believe if an organization achieves a higher CMMI maturity level, a higher performance of the software processes follows. The performance they achieve depends on their executions of each CMMI Process Area (PA). CMMI is a framework that helps improve software product quality and productivity but not processes. CMMI describes the characteristics of processes but not the processes themselves. In other words, CMMI just includes “what.” There have been difficulties in increasing productivity with these models because “how” is not within the scope of the CMMI. Thus, software engineers and project managers know the goal of their project, but they do not know how to implement each procedure of the CMMI PAs or have the means to improve the processes for their goal. SEI has introduced “how to” technologies for CMMI at the individual and team level with PSP and TSP. There are also several “how to” technologies, appraisal methods, PSP/TSP, and measurement and analysis tools, which are foundations of the solution for high performance of software process [3].

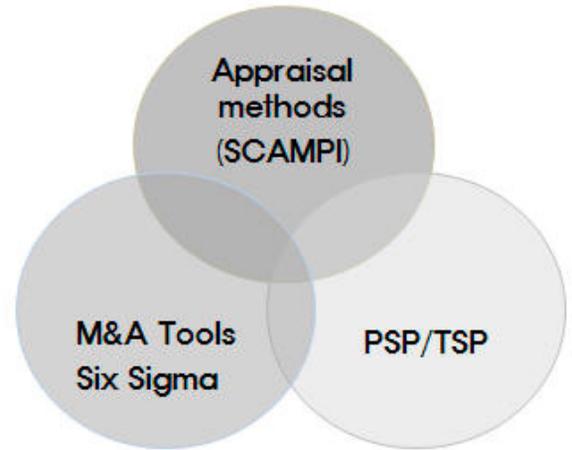


Figure 1: The “How to” Technologies for high performance

In this article, we focus on the integrated use of Six Sigma and PSP/TSP for higher performance of software processes and project management. PSP helps individual developers improve their performance by bringing a discipline to the way that they develop their software [4]. TSP provides software engineers and managers with a way to establish and manage their team to produce high quality software within a given schedule and budget [5]. Six Sigma is a quality improvement approach to enhancing an organization’s performance by using statistical analytic techniques [6]. It provides the quantitative analysis tools necessary to control process performance.

Many organizations that endeavor to improve software processes often find themselves integrating many approaches to achieve that improvement. Integrating Six Sigma and PSP/TSP can enable software engineers and project managers to analyze PSP/TSP data and to systematically improve process performance at an organization level. To do this, we map Six Sigma tools to each PSP/TSP process in order to show what Six Sigma techniques can be applied to the data of a given PSP/TSP and suggest Six Sigma practical usage guidelines to support process improvement activities at an individual and team level. However, there are a few analysis tools, such as process dashboard [7], Hackstat [8, 9], and PSP Assistant [10], and systematic process control functionality metrics collected in PSP/TSP activities. This article suggests Six Sigma and PSP/TSP tools that we have developed and proposes a framework for integrating those tools based on a knowledge-base repository. Thus, we can create a quantitative project management methodology by integrating Six Sigma and PSP/TSP based on a knowledge-base repository.

## Integrating the “How to” Technologies

Six Sigma supports software process improvement in PSP/TSP activities and helps organizations to achieve process improvement goals at an organizational level. PSP provides individual level project data containing information from what should be done to what has been done. Then, TSP is used in order to extend the collected data into a team view. Despite the fact that TSP activities are mostly based on the PSP results, PSP and TSP should be handled in different ways according to their different points of view. This means that the way of utilizing Six Sigma for each process should be different.

Since most of the PSP metrics and quality assurance activities are embedded in TSP, an adequate understanding of PSP is necessary. Six Sigma provides PSP/TSP with various tools for detecting special causes of variation, evaluating the impact of process changes, and improving process performance at an organizational level. Figure 2 presents the relationships among the “how to” techniques at an organization level. Using Six Sigma tools, individual-level data gathered from developers, following PSP0 through PSP3, is managed and analyzed. In addition, based on the PSP data, which is transformed into TSP data at a team level, TSP establishes a defined process foundation and generates useful data that can be analyzed using Six Sigma tools. The analysis results from Six Sigma provide methods for analyzing collected data in PSP/TSP and leads to individual and team level (further, organizational level) performance improvement through effective decision making.

Six Sigma provides various statistical and non-statistical tools in order to support effective decision making in the process of developing software. Mapping Six Sigma tools to each PSP/TSP process helps software engineers and managers understand how to use Six Sigma analysis techniques in conjunction with PSP/TSP data. Mapping the Six Sigma and PSP/TSP process, shown in Table 1, describes the statistical analysis and decision-making support tools of each PSP/TSP phase and its purpose. Since PSP/TSP activities are performed in several cycles, more Six Sigma tools can be applied in later cycles. The main issues of the mapping between Six Sigma and TSP actually rely on information gathered from the PSP activities. In other words, it is important to define what and how to extend the individual data to the team level information. To do so, we have defined several steps by tailoring the TSP launch process in [5]: Strategy, plan, risk, assessment, review and postmortem.

### An Integrated Framework for Six Sigma and PSP/TSP Six Sigma and PSP/TSP Tools

We suggest frameworks and implemented relative tools of Six Sigma and PSP/TSP, Six Sigma Project Management Tool (SSPMT), JASMINE, and ALADDIN, respectively [11, 12]. SSPMT is a web-based Six Sigma project management supporting tool that supports Define, Measure, Analyze, Improve, and Control (DMAIC) and Design for Six Sigma methodologies. Using the project initiation data and PSP/TSP data gathered from JASMINE and ALADDIN, SSPMT performs each step of DMAIC and provides analytic results. JASMINE and ALADDIN are web-based PSP and TSP project supporting tools, respectively. Since TSP mostly gathers information from the PSP activity results, most of the process works are done by using JASMINE. JASMINE collects an individual developer’s work product information such as Source Lines of Code (SLOC), fault counts, and so on. When a system is developed using Eclipse, it provides plug-in that automatically collects bug occurrence information per compiler. ALADDIN recollects the individual level project data and categorizes it at the predefined team level for further organizational decision making. Although the tools interact with each other, since they use individual data repositories, they are not fully integrated from the management point of view. Our intuition is that the decentralized database reduces the capability of managing the output of each process and further

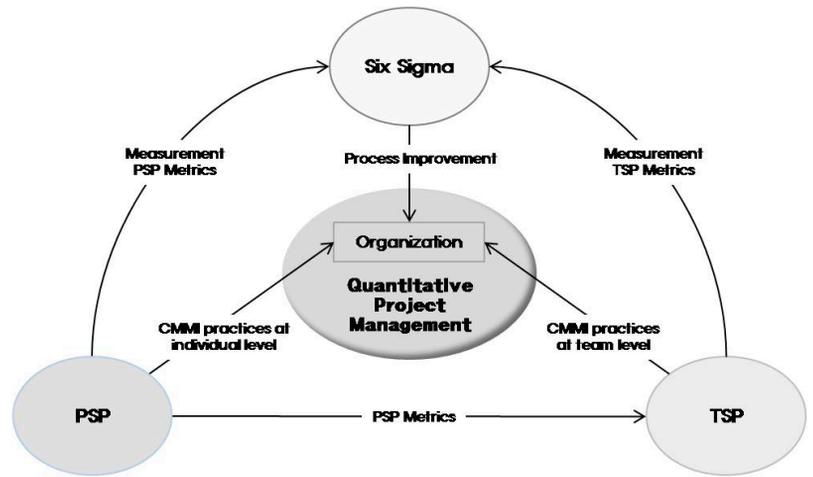


Figure 2: Integrating the “How to” technologies

TSP	PSP	Six Sigma Tools	Purpose	
Strategy	Project objective and strategy	Planning	Correlation analysis, Regression analysis	To estimate development time of new program
	Team Goals and Roles	Development	Dotplot, pareto analysis	To analyze the percentage of defects by defect type
Plan	Overall Plan	Postmortem	Correlation analysis, Regression analysis	To estimate size and development time of new program
	Quality Plan	PSP0.1 Planning	Correlation analysis, Regression analysis	To estimate Added and Modified value for new program
Risk Assessment	Project Risk Analysis	Development	Multiple regression analysis	To analyze Added, Reused, Modified value distribution of current program and to estimate LOC for new program
		Postmortem	The same tools as in postmortem phase in PSP0	
Review	Report Generation	PSP1 Planning	PROBE(PROxy-Based Estimating)	To estimate new program size and development time
	Management Review	Development	The same tools as in development phase in PSP0	
		Postmortem	2-sample t-test	To determine whether estimations has been correct by comparing between Plan Size/Hour values and To Date Size/Hour values
Postmortem	Launch Postmortem	...	Process Charter Process Evaluation Sigma Calculator	
	Quantitative Project Management	...		To estimate using quantitative indicator

Table 1: Mapping table of Six Sigma and PSP/TSP

quantifying decision-making variables or measurements.

As the PSP/TSP process continues, Six Sigma quantifies the results of the processes by using various tools in order to provide decision-making support. The detailed procedure of this process, using the existing tools (SSPMT, JASMINE, and ALADDIN), can be described as follows:

1. Initiate the PSP/TSP process using JASMINE and ALADDIN.
2. Store the PSP/TSP data in the data repository of SSPMT.
3. Analyze the PSP data and report individual level process performance improvement and decision-making issues using the SSPMT.
4. Organize the PSP data into the predefined team in order to support TSP data analysis.

5. Analyze the TSP data and report the team level process performance improvement and decision-making issues using the SSPMT.
6. Generate the results of both the individual and the team level decision-making report.
7. Keep track of and provide feedback to the next cycle.

First, the PSP and TSP are performed by following their own process using JASMINE and ALADDIN. Then, the overall data is stored in the data repository, which is in the SSPMT database. (We will discuss the repository in the next section in detail.) ALADDIN collects the results, gathered from JASMINE, and combines them in the form of teams that were predefined in TSP team building in order to support team level decision makings and analyses. Finally, the SSPMT generates the decision-making report according to the data and its analysis.

According to the procedures above and the mapping table shown in the previous section, the suggested architecture supports not only PSP/TSP activities, but also their relative analysis results and decision-making issues at an individual level and a team level. In addition, the results of the statistical data analysis help the project managers and software engineers to readily make various decisions, for example, changing the management. It is also easier to manage each process's data concurrently by integrating the data repositories.

However, since the tools use individually distributed data repositories, the measurements, relative matrix, and results are not managed concurrently and there is also needless storage waste. For example, since most of the data analysis results of the TSP are based on the PSP data, the TSP tool itself does not need to be inputted again and/or the data restored. Thus, it is better to directly store the necessary measurements and minimize duplication.

### Knowledge-based Data Repository

According to the facts stated above, the suggested architecture is based on the integration of the database of the three methodologies. Since it manages the overall data of each process results by integrating the database, a more quantitative and integrated process and project management can be provided.

In order to integrate the supporting tools of the three processes, we provide a knowledge-based database. As shown in Figure 4, the database architecture consists of three data repositories as follows:

1. **Master Data Repository:**  
 Project Master Data: contains project initiation data, such as baseline, team members, resources, schedule, measurements, process mapping information, etc.  
 Process Master Data: contains setup information for Six Sigma and PSP/TSP processes (e.g., DMAIC, DFSS of Six Sigma).
2. **Instance Data Repository:** stores each process's empirical data (measurement) produced by each tool (e.g., Six Sigma instance data, PSP/TSP instance data).
3. **Analytic Data Repository:** stores analytic results of instance data using Six Sigma data analysis.

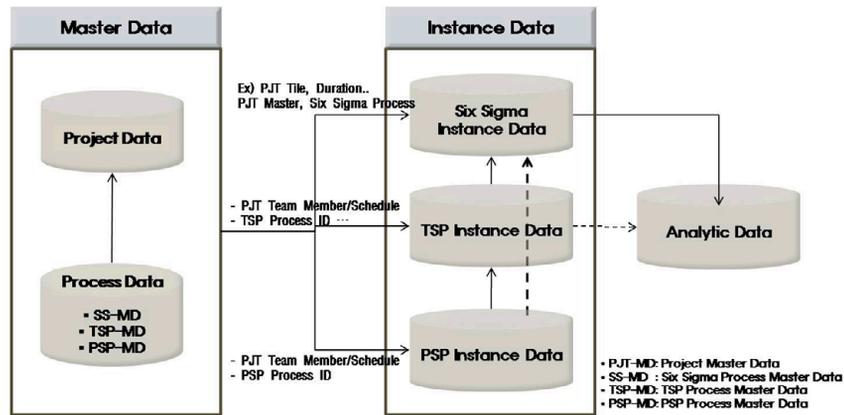


Figure 3: Data Repositories

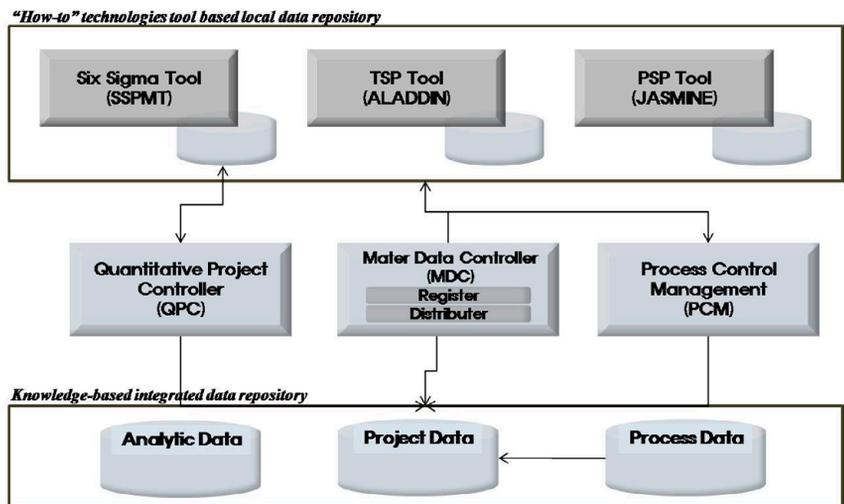


Figure 4: Integrated framework for data management

The master data repository, which contains project initiation data and process setups, provides the basic information of a project and its process to the instance data repository. For example, the Six Sigma framework and team information can be used in Six Sigma instance data and TSP data, such as process ID and process name, and PSP data, such as individual member information, PSP process ID, and name, can be used in TSP instance data, and PSP instance data, respectively.

Based on the data collected, PSP activities are performed and TSP extends the results to a team level using the team member information and process ID. While PSP/TSP tools perform their process, Six Sigma instance data repository collects the results in order to perform the data analysis that is used to support decision making. Finally, the analytic results are stored in the analytic data repository.

### Data Management

In order to support the data repository framework, we implemented a central data management application, QPC, MDC, and PCM for managing and analyzing the metrics from the PSP/TSP.

The register in the MDC, which is the central data management application, receives the process and project master data (e.g., project title, budget, schedule, and so on) from the managers. The distributor then distributes the information to each tool.

According to the data, JASMINE performs the PSP activities and ALADDIN combines the results at a team level and performs the TSP activities. The SSPMT then analyzes the measurements collected from the tools and provides decision-making reports. During the Six Sigma analysis, the PCM continually monitors the processes and feeds back the analysis results to the organization in the form of a report or e-mail. QPC transfers the analysis data of the SSPMT to the analytic data repository, and if required, to other data repositories.

### Supporting Decision Making An Integrated Process

In this section, we suggest a practical guideline showing how the framework can be used by providing an example process of the suggested framework. Figure 5 shows the overall process of the suggested framework. The process can be categorized into three layers: administrative, project, and organization.

First of all, the administrative layer works from the administrator's point of view. In this layer, the project manager registers the project basic information and its process information and maps the process and the project. As mentioned in the previous subsection, all the project initiation information can be registered with the MDC. If a similar project exists, the project manager revises it and uses a new project template.

In the project layer, PSP and TSP processes are performed according to the predefined mapping information. As shown in Figure 5, each project team member first performs his or her individual role by following the PSP activities PSP0 through PSP3. Then, they gather the individually performed outputs (e.g., SLOC, fault count) in order to extend it to the team level using the TSP based on the team information and TSP process ID gathered from the administrative layer. Data collected in this layer is stored in the instance data repository.

In the organization layer, using the individual and team level information gathered from the previous step, the Six Sigma process is performed using appropriate tools based on the predefined framework. Using statistical and non-statistical analysis, Six Sigma provides analysis results at the project, individual, team, and organizational levels. According to the results, the SSPMT provides an analysis report that quantifies the overall results and enhances organizational decision making. By using the PCM, project managers can also monitor whether each project is going well. Finally, the feedback based on the overall results and relative reports can be used to improve the whole development lifecycle and further organizational improvement.

### Quantitative Project Management

Based on the suggested integrated framework and processes, we can collect individual ( $x_{psp}$ ) and team ( $x_{tsp}$ ) data through the PSP/TSP processes at the project layer. We can also elicit a set of metrics ( $x_{tsp'}$ ) from individual ( $x_{psp}$ ) and team ( $x_{tsp}$ ) data in the TSP phase. The data is analyzed using Six Sigma tools at each project layer and organization layer. Then, the Quantitative Man-

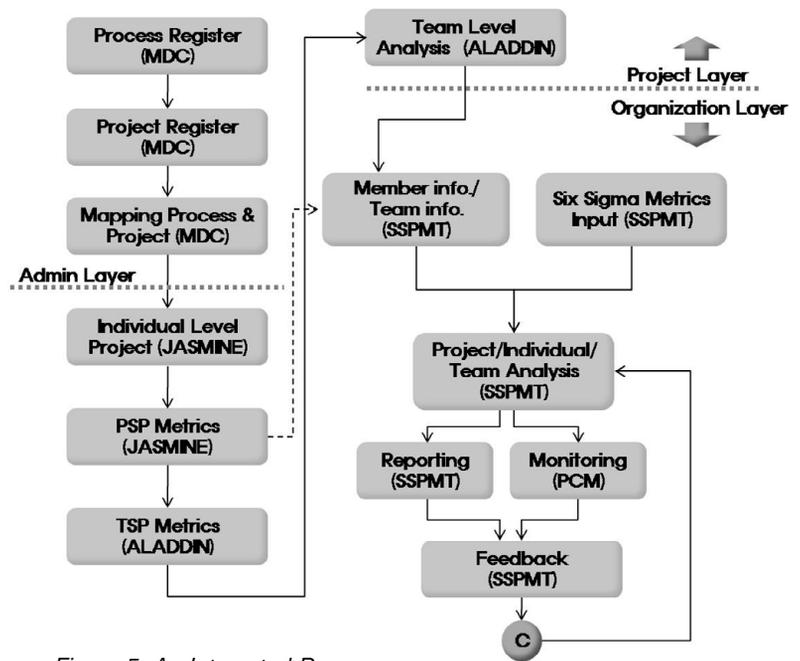


Figure 5: An Integrated Process

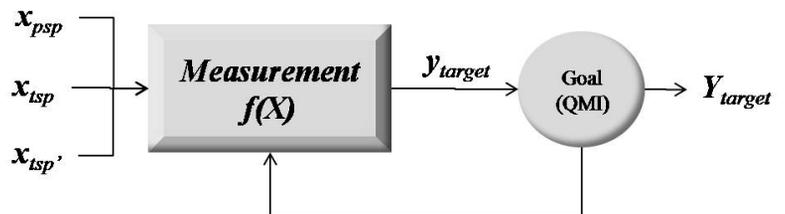


Figure 6: Quantitative Project Controller

agement Indicator (QMI) absorbs the analyzed data ( $y_{target}$ ) and determines if it satisfies the organizational goal. If so, the process continues the same as at present. If not, individuals and the team will receive feedback indicating what and where the problems are. Then, the process will be changed or fixed according to the issues and the process will be repeated with newly collected metrics until the QMI confirms that the goal is satisfied.

Using the QMI, it is possible to directly relate the individual/team level data and related metrics to the organizational goal. Since it indicates the locations of the cause of the disconfirmation, organizations can reduce the cost and change the schedule of the process execution. As a result, by applying the QMI at each PSP/TSP phase within Six Sigma's quantitative measurements, organizations can deliver their products with the desired quality, which will lead to customer satisfaction.

### Conclusion

This article focused on supporting quantitative decision making for process performance during software development projects. It is proposed to seamlessly integrate Six Sigma and PSP/TSP tools using a knowledge database. Thus, an organization can continuously improve its process based on empirical and analytic data and move to a higher CMMI level. In the future, we expect to develop more accurate metrics for quantitative project management of each domain and project guidance. ♦

## Acknowledgements

This research was supported by The Ministry of Knowledge Economy, Korea, under the ITRC(Information Technology Research Center) support program supervised by the NIPA(National IT Industry Promotion Agency). (NIPA-2009-(C1090-0902-0032))

## Disclaimer

<sup>®</sup>CMMI and <sup>®</sup>CMM are registered in the U.S. Patent and Trademark Office by Carnegie Mellon University.

<sup>SM</sup>PSP and <sup>SM</sup>TSP are service marks of Carnegie Mellon University.

## REFERENCES

1. Mark C. Paulk, B. Curtis, M. B. Chrissis and C. V. Weber, "Capability Maturity Model for Software, Version 1.1", Technical Report CMU/SEI-93-TR-024, ESC-TR-93-177, Feb, 1993.
2. CMMI, "Capability Maturity Model Integration" Software Engineering Institute, CMU, 2002.
3. CMMI Made Practical, "Implementing CMMI for High-Performance", London, April, 2009.
4. Watts S. Humphrey, "The Personal Software Process", Technical Report, CMU/SEI-2000-TR-022, ESC-TR-2000-022, Nov, 2000.
5. Watts S. Humphrey, "The Team Software Process", Technical Report, CMU/SEI-2000-TR-023, ESC-TR-2000-023, Nov, 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. <<http://www.processdash.com/>>
8. Johnson, Philip M., "Project Hackstat: Accelerating adoption of empirically guided software development through non-disruptive, developer-centric, in-process data collection and analysis", Technical Report csdl2-01-13, Department of Information and Computer Sciences, University of Hawaii, Honolulu, Hawaii 96822, November 2001
9. <<http://code.google.com/p/hackystat/>>
10. R. Sison, et al, "Personal Software Process (PSP) Assistant", In Proceedings of the 12th Asia-Pacific Software Engineering Conference (APSEC'05)
11. Y. Park, H. Park, H. Choi, and J. Baik, "A Study on the Application of Six Sigma Tools to PSP/TSP for Process Improvement", In Proceedings of the 5th IEEE/ACIS International Conference on Computer and Information Science and 1st IEEE/ACIS International Workshop on Component-Based Software Engineering, Software Architecture and Reuse (ICIS-COMSTAR'06).
12. P. Zedan, H. Park, and J. Baik, "A Six Sigma Framework for Software Process Improvements and Its Implementation", In Proceedings of APSEC 2007, Dec 5-7, 2007.

## ABOUT THE AUTHORS



**Sejun Kim** is an Engineer at BISTel, Seoul, Korea. Kim earned a master's degree in computer science at the Korea Advanced Institute of Science and Technology (KAIST) and a bachelor's degree in computer science at Kwangwoon University. Kim was involved in the Software Process Improvement Center project and has developed the SSPMT, CMMI-Six Sigma Guideline, and Reliability-Six Sigma Guideline. His research interests are software process and quality improvement, especially focused on Software Six Sigma and Software Testing.

### KAIST

**335 Gwahak-ro (373-1 Guseong-dong), Yuseong-gu, Daejeon 305-701, Republic of Korea**

**Phone: +82-42-350-3356**

**E-mail: sejunkim@kaist.ac.kr**



**Okjoo Choi** is a research assistant professor at the department of computer science at KAIST. Choi earned a bachelor's degree, a master's, and Ph.D. in computer science from Sookmyung Women's University, Seoul, Korea. Before she joined KAIST, she had worked at ERP consulting services, Oracle Korea Ltd. as a consulting technical manager from 1996 to 2009 and LG Electronics Production Engineering Research Center an assistant research engineer from 1990 to 1996. She is currently involved in projects related to software reliability for embedded weapon systems.

### KAIST

**335 Gwahak-ro (373-1 Guseong-dong), Yuseong-gu, Daejeon 305-701, Republic of Korea**

**Phone: +82-42-350-3356**

**E-mail: okjoo.choi@kaist.ac.kr**



**Jongmoon Baik** is an associate professor at the department of computer science at KAIST. Baik earned a bachelor's degree in computer science and statistics from Chosun University and a Master's and Ph.D. degree in computer science from University of Southern California. Before he joined KAIST, he was a principal research scientist at Software and Systems Engineering Research Laboratory, Motorola Labs from 2001 to 2005. He is also an adjunct faculty member for the Masters of Software Engineering program at Carnegie Mellon University.

### KAIST

**335 Gwahak-ro (373-1 Guseong-dong), Yuseong-gu, Daejeon 305-701, Republic of Korea**

**Phone: +82-42-350-3356**

**E-mail: jbaik@kaist.ac.kr**