International Journal of Reviews in Computing
10th April 2012. Vol. 9
IJRIC

ISSN: 2076-3328 www.ijric.org E-ISSN: 2076-3336

A LITERATURE REVIEW & RECOMMENDATIONS ON PERSONAL SOFTWARE PROCESS TOOLS

WAQAR UL HASNAIN, SHAHBAZ AHMED
Reserch Scholar, Department of Computer Science and Software Engineering, International Islamic University, Islamabad, PAKISTAN.
Asst. Professor, Department of Computer Science and Software Engineering, International Islamic University, Islamabad, PAKISTAN.

E-mail : waqarulhk@gmail.com, shahbaz.ahmed@iiu.edu.pk

ABSTRACT

Review papers and Survey research papers are the primary source of information that helps researcher to find updated knowledge in that domain. No systematic survey or review paper was found in the field of Personal Software Process Tools and Software Architecture Evaluation Tools. This requires the need of performing the literature review. The review conducted tries to fill the gap in the literature and recommends some suggestions for tool that support integrated Personal Software Process and Architecture Evaluation Process. Such tools will not only help individual software engineer to perform single-case holistic design case studies which are required to fill the gap in the current literature, but also help in commercial environment for reducing cost, and schedule.

Keywords: *Personal Software Process (PSP), Tool Support, Software Architecture, Architecture Evaluation, Cost, Schedule*

1- INTRODUCTION

1.1- Personal Software Process.

The Personal Software Process (PSP) [1] is the initiative of Software Engineering Institute (SEI) developed to improve the performance of individual software engineer especially for small organizations. It is a structured, disciplined, and measureable software process for individual engineers. The PSP improve quality and productivity of engineers and make their result more predictable. Using PSP engineers can identify areas where they can improve from their process feedback. An individual engineer can use PSP principles in any of 21 key process areas of CMM/CMMI-DEV-1.2 [4]. However, initial work reported focuses only on design, code, and test phases. One of the principles that drive the PSP is [1]:

"With this knowledge, the engineers can select those methods and practices that best suit their particular tasks and abilities."

From this principle it is clear that software engineers can use new appropriate methods within PSP to improve their quality of their work. The process phase where the quality is more appropriately addressed is software architecture [6]. Many software architecture design and evaluation method, techniques, models and processes have been proposed such as ATAM/CBAM, and SARA. These methods will be discussed in next section.

The PSP is designed for small organizations and for individual engineers and does not define team software processes. However, PSP engineers can be guided through a team oriented approach i.e. Team Software Process (TSP). PSP does not depends on the use of any special tool, but PSP engineers have to go through many phases and have to fill out many forms and prepare script which utilizes huge development time. Therefore, automation of PSP will reduce development time and cost. The automation will also help in process learning as indicated in [1]

"With CASE facilities to automatically log time, track defects, maintain data, and present statistical analyses, the PSP likely would be easier to learn and more efficient to use."

Various research and academic PSP tools are available some of these tools are briefly explained in Section-3.

The PSP has successful track of industrial application especially in DEC, HP and AIS [1]. Another publication [2] reports the use of PSP in Motorola Paging Products Group, Union Switch & Signal Inc. and Advanced Information Services Inc. The most projects executed at these organizations contain one to three software engineers. However,

ISSN: 2076-3328 www.ijric.org E-ISSN: 2076-3336

project executed at AIS involved two groups with three to five software engineers. The most projects executed at Motorola contain zero defect which exposes the significance of using PSP in industry. The current version of PSPBOK [3] explains the seven necessary competency areas: Fundamental Knowledge; Basic PSP Concepts; Size Measuring and Estimating; Making and Tracking Project Plans; Planning and Tracking Software Quality; Software Design; Process Extensions and Customization. However, it still does not address architecture design and evaluation competency.

1.2- Software Architecture Design and Evaluation.

There are many software architecture design and evaluation processes, brief survey of these methods can be found in [7]. The most used evaluation methods are SAAM, ATAM/CBAM, and SARA. ATAM is considered to be standard method of software architecture evaluation within CMMI organizations. ATAM/CBAM along with Quality Attribute Workshop (QAW); Attribute Driven Design (ADD); View and Beyond (V&B); and Active Reviews for Intermediate Design (ARID) form a systematic process and is called Architecture-Centric Engineering (ACE). The ACE is currently integrated in Team Software Process (TSP) which provides an accelerated process with focus on quality early in the architecture phase [5]. However, the current implementation of architecture evaluation method is for TSP coaches or managers. The architecture evaluation method should be one of the activities of software engineers or developer as indicated in [6]

> *"It has convinced management that developers need architectural analysis up front."*

2- RESEARCH STRATEGY

For performing literature review guidance of [19] was used. The research strategy consists of formation of research question; selection of appropriate keywords; formation of search string; selection of research resources; and formation of inclusion and exclusion criteria. All of these are discussed one by one.

Research Question:

Which tools are developed for PSP with Software Architecture Design and Evaluation support?

Keywords:

The keywords selected from research question are: "Personal"; "Software"; "Process"; "Architecture"; "Evaluation"; "Tools".

Search String:

The search string was formulated based on keyword. Following search string were used.

1- Software Architecture Evaluation Tools.
2- Personal Software Process Tools.

Search Sources:

The search sources include digital libraries, publishers, search engines, and references/bibliography. These sources include:

ACM Digital Library; IEEExplore Digital Library; ScienceDirect; Springer Verlag; Springer Science + Business Media; Kluwer Academic Publishers; Elsevier; Wiley InterScience; IEEE Computer Society; British Computer Society; Web of Science; CiteSeer; and Google.

Phase-1: The reviewed articles include research papers from Conferences, Journals, Workshops, and Symposiums. Copy rights Technical Reports, Doctoral Dissertation with ISBN, ISSN or Copy rights. However, White Papers, Technical Reports, Website Articles without copy rights were not considered. Total selected publications for primary study using search string= 36. **Phase-2:** Total selected research papers for primary study for detailed study= 36. **Phase-3:** Total selected research papers before applying inclusion and exclusion criteria= 22. **Phase-4: Inclusion Criteria:**

All publication related to: Personal Software Process Tools; Software Architecture Evaluation Tools; Software Process Commercial Tools;

Exclusion Criteria:

All publication not related to PSP tools, architecture evaluation tools or quality management tools.

However, due to space limitation only small sample of publications are selected.

Total selected research papers after applying inclusion and exclusion criteria=11

3- PERSONAL SOFTWARE PROCESS TOOLS

PSP-DROPS

PSP Data RepOsitory and Presentation System (PSP-DROPS) is a web based tool [8] that support PSP. The tool is developed at Embry-Riddle Aeronautical University to automate the PSP process and to facilitate the teaching of PSP. PSP-DROPS helped in cut down the work load of management and analysis of PSP data.

PSP-DROP provides assistance in filling different form, performs calculation on collected data, store process data in database, and generates graphical analysis. These reports and analysis result can be generated anywhere in the world. The web base architecture of PSP-DROP fulfills the goal of its

ISSN: 2076-3328 www.ijric.org E-ISSN: 2076-3336

development for PSP data for academic and industrial use.

The four major software architectural component of the PSP-DROP are PSP database server, CS1 module, CS2 module and a Teaching Assistant module (TA). These components communicate with HTTP Server and end user communicates with system through Web Browser.

PSP Database Server was implemented using rational database for storing the Personal Software Process data entered by the computer science student. This PSP Database Server is connected with the University database and retrieves data automatically.

The security features of PSP-DROPS allow only authorized and authenticated user to login and retrieve his own PSP data. However, student can only enter data and are not allowed to update the data.

Hackystat:

Hackystat is a tool [9] that supports Personal Software Process for collecting metrics regarding effort, size, and defects. It is developed at University of Hawaii and used by its students. The basic purpose of development of this tool is to reduce "context switch" between process recording and product development. This tool also collects metrics regarding the development activities of software engineer. Size data is collected using C.Kemerer Object Oriented metrics for .class file of Java. Defect data for pre-release and post-release is automatically collected through attached sensors to JUnit and Bugzilla respectively. The tool comes with Hackystat server which has ability to create user with password. Different metrics are collected using sensors and are sent to server for historical data collection. The server sends email alerts to user for their activities. These alerts provide a just-in-time approach for metrics collection and analysis or results. The server is written in Java language which contains about 200 classes. It provides client-side sensors for attachment to development tools. These sensors have ability of automatically collecting PSP data and provide automatic calculation of effort, size, and defects. However, Hackystat does not support all types of derived metrics and analysis required in PSP e.g. cost of removal of a defect. There is other limitation such as Hackystat focus only on coding activity [10].

PROM:

PRO Metrics (PROM) data collection and analysis tool [10] is developed for Personal Software Process. PROM provides automated data collection and analysis facility for both code and PSP process measures. This data not only contains PSP data but also procedural metrics, object oriented metrics, and custom metrics. PROM provides data collection and analysis facilities for personal, workgroup and at enterprise levels. The architecture of PROM is based on plug-in technology and use (SOAP) to communicate with various component and subsystems of the architecture. This architecture based on Package-Oriented Programming that makes the development and integration of its components easier and extensible. PROM is consists of four component Database, PROM Server, Plug-in Server and Plug-in. Database of PROM is used to store data regarding PSP data, software metrics and project activities. PROM Server use SOAP web services to communicate with other components. Plug-in Server collect data from plug-ins provide caching facility and communicate with PROM Server for data storage. The fourth component is Plugs-in for IDE, these Plug-ins communicate with PROM Server using SOAP protocol. PROM not only provides metrics and process support to developer but also to the manager. Developers can simultaneously login for pair programming and can access PSP data, and software metrics for analysis and improvement. PROM is fully automated to support the context switching problem in process recording and product development. It also helps in collection of data regarding Activity-Based Costing (ABC) to manage cost of project for project manager. PROM resolves the privacy issues of PSP data analysis i.e. manager cannot access private data of individual software engineer. However, manager can only access data that is relevant to project monitoring and control. PROM is developed using Java technology and its components communicates using XML and SOAP. However, PROM also supports manual data insertion facilities.

Personal Software Process Assistant: PSPA

Personal Software Process Assistant (PSPA) [11] designed and developed to facilitate the automatic collection of PSP data, viewing and editing PSP logs and reports. The tool also performs automatic classification and ranking of defects. The tool has the capabilities of recording compile defects of programs written in Java and C. It not only automatically collects these defects but also classifies for individual software engineer and for team. PSPA is written in .Net(C#) with plug-in support written in Java. It uses centralized local database.

ISSN: 2076-3328 www.ijric.org E-ISSN: 2076-3336

PSPA provides daily schedule tracking facilities, automatic Line of Code counting facility, compile defects recording facility, size and time estimation facility, Time and Defect Recording Logs facility, and facility of automatic classification and ranking of defects. The data related to these facilities is collected through plug-ins and stored in a local centralized database. These plug-ins communicate with Eclipse open source IDE. PSPA provides tracking individual engineer task with the help of Gantt chart and a timer attach to it. Timer is invoked automatically when an engineer starts working PSPA tools. However, if software engineer forget to start timer it prompt for its operation. The PSPA provides support for manual data entry regarding interruption time. It provides planning and scheduling facility in standard PSP formats.

PSP engineer and project manager both can take advantage of PSPA tool. Project manager can create project, decompose into small tasks, and assign these task to developers. Software developer can also create private tasks. However, private tasks can only be viewed by the developer and project manager will not have access to them.

PSPA provides automatic defect recording and classification through plug-ins. Time is automatically calculated using timer and logged in the database. These basic measures are used to calculate defect density and productivity of individual engineer. Defects are logged and classified using standard PSP Defect Type Standard. These defects are ranked based on the frequency of their occurrence.

PSP logs and reports are automatically produced, these reports includes Productivity per Task; Yield per Task; Defect Density per Task; Estimate Size versus Actual Size; Estimated Time versus Actual Time; Estimated Cost versus Actual Cost; Productivity of Team; and History of Member. The consolidated team Gantt chart provides information regarding individual and team productivity. This information is only available to the manager for project monitoring and control. The defects information provides the quality at individual and at the team level.

DuoTracker:

DuoTracker is software tool [12] designed and developed to address the process tracking and analysis needs of individual software engineer and for organization. Its defect classification capabilities and data collection is based on ISO 9001 and CMM standard. The tool is developed to address the issues related to adoption barrier to PSP tool such as: manual entry of data; switching between applications; and collection of rigid data. DuoTracker provides the solution to this problem by integrating PSP data collection tool in ISO 9001 and CMM based defect tracking tool. The eleven mandatory categories of IEEE Standard 1044-1993 was used for defect classification scheme and implemented in CMM based software lifecycle. These defects are collected using defect logger implemented as one of the component of DueTracker. DueTracker has ability of integrating with organization wide defect tracking and analysis tool. Software engineer can select any type of defect for his own analysis for the available defect type in this tool. Defect classified in IEEE standard 1044 are automatically logged whereas some PSP specific and unique field are manually entered. DueTracker provide a facility for comparing individual PSP quality data with organization wide product quality data. The tool provides a time logging facilities for each defect, such as the time at which defect was found and fixed. It allows engineer to analyze estimated defect fix time as well as actual fix time. However, in DueTracker recording of compilation errors are not automated. Another issue with the tool is security of PSP data which is restricted from viewing using some measures. The defect records in the DuoTracker system is viewed by Defect Viewer. This viewer provides the necessary information regarding defect classification, logged-on user, and project. DueTracker provides two different ways of updating defects one is for assigned defects and other is for unassigned defects.

PSP-Expert Visualization Agent: PSP-EVA

PSP-Expert Visualization Agent shortly PSP-EVA [13] provides automated support of Personal Software Process (PSP) for software engineers. It covers all of the phases of PSP i.e. from PSP0.1 to PSP3.0. The tool is designed using agent-oriented concept, these agents provides personal assistance for software engineer. The agents used in the tool are InterfaceAgent (IA), TaskAgent (TA) and SearchAgent (SA). These agents are introduced in the PSP-EVA to perform tasks such as simplifying plan and schedule. It provides visual representations of defect density and project progress. The tool has a capability of multi-tasking. PSP-EVA tool provides a visualization support for performance of software engineer. PSP engineer and Project Manager can have access the personal data for monitoring and evaluating performance. However, Project Manager can only view and cannot change personal data. PSP-EVA provides a role base login

ISSN: 2076-3328 www.ijric.org E-ISSN: 2076-3336

support for users such as software engineer or a manager.

Sensor-Based Scheduling:
PSP-EVA provides a sensor based scheduling and tracking of work progress. It has the ability of automatically store data for historical analysis. For scheduling Gantt chart is provided in the tool for graphical representation of milestone. This visual representation of milestone in Gantt chart is accomplished with the help of agent that enhanced PSP tool.

Performance Visualization:
Performance visualization is accomplished with the help of interface agent. The user can monitor his performance with the help of these interface agents. These agents send alert messages to software engineers when they are slow.
Finally the PSP-EVA provides analysis support for both software engineers and for project manager.

PSP Tool JTemporalAPI
The tool discussed in [14] support Personal Software Process with focus on problem of context switching and recording overhead faced by the users while recording time log. The tool used the speech sensor in order to record activities. The tool provides the facilities to record start and end time during the particular activity. Microsoft Speech recognition API was used in the tool to recognize the speech of user for recording start and end time. However, this approach had two problems; first it is unable to record activity time if user forgets to record it; second it is unable to identify the start and end time. Activity duration estimation was performed using additional sourced of information such as information automatically obtained from secondary sensor and information obtained from schedule stored in database. The tool uses sixty rules and seven algorithms to record start and end time of an activity automatically.

4- ARCHITECTURE EVALUATION TOOLS

DSA Documentation System:
DSA Documentation System [15] (shortly DDS) is a web based software architecture documentation management tool with features such as configuration management system. The DDS provides security features such as authorization and authentication for each stakeholder. The access rights are controlled by password protected login and user can only login with assigned role. The major components of the DDS tool are: **General Information:** General information is concerned with software architecture information. **Form:** It provides capabilities such as cut, past or files submission. **Tip:** Tips are used to provide help to the user in the form of mini Flash diagrams tutorials. **Glossary:** Each page of DDS tool provides glossary associated with the document. **Acronym List:** Acronyms are included in each document with associated links. DDS consists of four major sub-systems which are explained below:

DDS Role-Based Access Management System:
DDS role based access management feature provides varying degrees of access to different sections of the tool. Only authorized person can access the required specific portion of the architecture. Different roles can be assigned such as architect, junior architect, documenter or manager. Each role assigned to the user of DDS comes with a set of tips whose contents depends base on role of user.

DDS File Management System:
The file management is performed using web forms. All information typed in web forms are uploaded as an ASCII text files. Web forms also provide features such as cut and paste. All files that were generated previously are uploaded as Multipurpose Internet Mail Extensions (MIME) file types.
The output of the file management depends upon stakeholder need and role. The available output forms are: Filled-in Forms; PDF forms; and Template based PDF forms.

DDS Configuration Management System:
DDS configuration management system provides role based access with configuration management back end which allow variety of configurations at any time. The configuration management system capability allows the architect to change the document with configuration id.

DDS Extensibility and Configuration:
DDS tool is reconfigurable and extendable; an architect may configure it for new configuration item.
The configuration items includes stakeholder type, tips and forms where as extensible items include tips and forms.

Scenarios Based Architecture Evaluation Tool
The tool and its architecture discussed in [16] are for scenario based software architecture evaluation. It supports evaluation methods such as Software

ISSN: 2076-3328 www.ijric.org E-ISSN: 2076-3336

Architecture Evaluation Method (SAAM) or Architecture Level Modifiability Analysis (ALMA). The logical view of the architecture of the tool composed of following modules.

Scenario Elicitor: This module is used to record scenarios during scenario elicitation process. It communicates with a repository named "scenario classes" which contains predefined domain specific scenarios classes. This module also communicates with another repository named "checklists" which contains domain specific generalized questions.

Scenario Manager: Scenario manager module provides the service of organizing domain specific scenario repository. Sample scenarios can get from "scenario elicitor" module.

Scenario Classifier: scenario classifier module provides the facility of prioritizing scenario, classifying scenarios as direct or indirect scenario. The module also maintain list of such scenarios.

Architecture Representation Module: The module communicates with a repository named "Architecture Repository" for getting domain specific architecture samples. The module presents architecture documentation to architect and evaluator.

Evaluator: This module is designed to perform impact analysis for indirect scenarios. This activity is human intensive and evaluator module communicates with repository for getting any stored evaluation.

Result Presenter: The result presenter module communicated with Evaluator module and gets and presents results stored in "Result Templates" repository.

PAKME:

Process-centric Architecture Knowledge Management Environment (PAKME) [17] is a web based tool for managing software architecture knowledge and rationale. The web based architecture of PAKME provides architecture knowledge management facility for people involved in architecture process and located in various part of the world. The tool has developed using Hipergate which is an open source groupware platform. Hipergate provides various features such as project management and online collaboration tools. The purpose of this tool is to improve software architecture process. The tool supports the two categories of architecture knowledge management one is contextual and other is technical. The technical category of architecture knowledge concerning patterns, styles, tactics and analysis is covered in this tool. The tool provides a knowledge source of historical experience-based design decisions. The historical repository of knowledge provides guidance in making new decisions. The data model of PAKME provides two types of access one is organizational wide or generic knowledge regarding different projects. The other access is project specific or concrete knowledge regarding design history, architectural views and analysis results.

The user interface of the PAKME is developed using Java Server Pages and HTML technologies. These pages communicate with other sub-system through Hipergate. The user interface provides knowledge acquisition forms for capturing project specific and organizational specific knowledge. The major components of the PAKME are explained below.

Knowledge management component uses data management component for managing artifacts, store, retrieve, and update artifacts.

The Search component provides three types of search function such as keyword based search, navigation based search, and advanced search.

The Reporting component provides the service of generating various reports for architecture evaluation.

Finally the Repository Management component provides the service of storing data into the PostgreSQL 8.0 and retrieving data from it.

PAKME Services:

PAKMS provides various services which are discussed below.

Various forms and editing tools are provided through Knowledge Acquisition Service. These forms are used to enter project specific and organization specific knowledge into the database.

Different operations such as modification of data and deletion of data are performed using Knowledge Maintenance Service.

Knowledge Retrieval Service use different search functions to locate and retrieve architecture artifacts.

Finally, the Knowledge Presentation Service provides the service of presenting knowledge in a structured way.

5- COMMERCIAL TOOL

HP Quality Center:

HP Quality Center [18] is a web based quality management software solution. The tool enforces standardized processes across projects. It provides sharing and reusing asset libraries across projects. HP-QC is available in three versions: Starter

ISSN: 2076-3328 www.ijric.org E-ISSN: 2076-3336

Edition; Enterprise; and Premier. Based on the version of HP-QC it has different modules such as requirements management; release and cycle management; defect management; test plan; test lab; and dashboard reporting. All the information is stored in a central repository supported by either MS SQL Server or Oracle Database Server.

Different roles can be created in the tool and can assign different responsibilities with proper authentications and authorizations.

Requirement Management Module:

Central repository for managing requirements provides real-time visibility of their coverage and associated defects. The requirement management module provides multi-dimensional traceability between requirement, defects, test, releases and test cycles. The tool also provides facilities of prioritizing requirement based on business risk.

Test Plans Module:

This module provides facilities such as building test plans and designing test.

Release and Cycle Management Module:

The release and cycle management module is designed to manage software releases against plan.

Test Lab Module:

HP-QC can be used for unit, functional, load, regression and integration testing. The test lab module provides facilities such as to run scheduled tests unattended. The module also provides facility of test reuse.

Defect Management Module:

The defect management module can be used in defect detection, resolution or verification.

Versioning and Baselining Module:

Version control of requirements, business components, and tests scripts can be performed with this module. Baselining of these artifacts can be done at any strategic points in lifecycle.

6- OPEN RESEARCH PROBLEMS

As explained in Section-1 the PSP need its automation to reduce cost and time spent in manual work. However, there are some issues in current PSP process that also need to be addressed. The PSP needs to be integrated with architecture design and evaluation methods. This process integration will result in better management of cost, schedule, and project risks. However, no such process exist or tool that address integrated features of PSP, ATAM/CBAM, ADD, QAW, V&B, and ARID.

7- MISSING IN LITERATURE

Literature review and survey research papers are the primary and central source of information for any domain. However, no such survey or systematic review paper was found that addresses the need for PSP tool and its integration with architecture design and evaluation processes. The other most important thing is that no such tool exists that provides the integrated process support for PSP and ATAM/CBAM, ARID, or ADD. Such tool if available will fill the gap in literature, and will help in achieving competencies such as architecture design and evaluation. Such competencies if an engineer has will help in better estimation of cost, schedule, and risk identification early in the development lifecycle.

8- CONCLUSION

To reduce cost, time and risk in various phases of software development lifecycle a web based tool is required to manage basic SDLC phases in global software engineering. The process data should store in a central repository with 24/7 hours of availability for geographically distributed team.

9- FUTURE WORK

To address the current issues of PSP process and its available tools, a tool is under development stage. The tool will provide the integrated features of PSP and architecture design and evaluation methods.

REFERENCES:

[1] W.S. Humphrey, “The Personal Process in Software Engineering,” IEEE 1994.

[2] P. Ferguson, W.S. Humphrey, S. Khajenoori, S. Macke, A. Matvya, “Results of Applying the Personal Software Process,” IEEE 1997.

[3] M. Pomeroy, R. Cannon, T.A. Chick, J. Mullaney, W. Nichols, “The Personal Software Process (PSP) Body of Knowledge,” Version 2.0, (CMU/SEI-2009-SR-018), Carnegie Mellon University, Software Engineering Institute.

[4] CMMI Product Team, “CMMI for Development,” Version 1.2, (CMU/SEI-2006-TR-008), Carnegie Mellon University, Software Engineering Institute.

[5] R.L. Nord, J. McHale, F. Bachmann, “Combining Architecture-Centric Engineering with the Team Software Process,” (CMU/SEI-2010-TR-031), Carnegie Mellon University, Software Engineering Institute.

[6] R. Kazman, G. Abowd, L. Bass, and P. Clements, “Scenario-Based Analysis of Software Architecture,” *IEEE Software*, 13(6) pp.47-55, November 1996.

ISSN: 2076-3328 www.ijric.org E-ISSN: 2076-3336

[7] L. Dobrica, and E. Niemela, "A Survey on Software Architecture Analysis Methods," *IEEE Transactions on Software Engineering*, 28(7), pp.638-653, 2002

[8] I. Syu, A. Salimi, M. Towbidnejad, T. Hilburn, "A Web-Based System for Automating a Disciplined Personal Software Process (PSP)," IEEE 1997.

[9] P.M. Johnson, H. Kou, J. Agustin, C. Chan, C. Moore, J. Miglani, S. Zhen, W.E.J. Doane, "Beyond the Personal Software Process: Metrics collection and analysis for the differently disciplined," *Proceedings of the 25th International Conference on Software Engineering (ICSE'03)*, IEEE Computer Society.

[10] A. Sillitti, A. Janes, G. Succi, T. Vernazza, "Collecting, Integrating and Analyzing Software Metrics and Personal Software Process Data," *Proceedings of the 29th EUROMICRO Conference "New Waves in System Architecture" (EUROMICRO'03)*, IEEE Computer Society.

[11] R. Sison, D. Diaz, E. Lam, D. Navarro, J. Navarro, "Personal Software Process (PSP) Assistant," *Proceedings of the 12th Asia-Pacific Software Engineering Conference (APSEC'05)*, IEEE Computer Society.

[12] O. Akinwale, S. Dascalu, M. Karam, "DuoTracker: Tool Support for Software Defect Data Collection and Analysis," *Proceedings of the International Conference on Software Engineering Advances (ICSEA'06)*, IEEE Computer Society.

[13] H. Hassan, M.H.N.M. Nasir, S.S.M. Fauzi, "Incorporating Software Agents in Automated Personal Software Process (PSP) Tools," ISCIT 2009 IEEE.

[14] A. Ibrahim, H.J. Choi, "Activity time collection and analysis through temporal reasoning," ICACT 2009 IEEE.

[15] J.A. Stafford, "Creating and Using Software Architecture Documentation Using Web-Based Tool Support," (CMU/SEI-2004-TN-037), Software Engineering Institute, Carnegie Mellon University. September 2004.

[16] M. Usman, N. Ikram, "The Architecture of a Tool for Scenario-Based Software Architecture Evaluation," 2006 IEEE

[17] M.A. Babar, I. Gorton, "A Tool for Managing Software Architecture Knowledge," *29th International Conference on Software Engineering Workshops (ICSEW'07)*, IEEE Computer Society.

[18] Hewlett-Packard Development Company, L.P., "HP Quality Center software," (4AA0-9587ENW) Rev.3, Data sheet, February 2009.

[19] Keele University and University of Durham, UK, "*Guidelines for performing Systematic Literature Reviews in Software Engineering,*" (EBSE-2007-01), Version 2.3, 2007.