A Multi-Process Quality Model: Identification of Key Processes in the Integration Approach

Aedah Abd Rahman[1], Shamsul Sahibuddin[2] and Suhaimi Ibrahim[3]

[1]Software Engineering Section, MIIT, Universiti Kuala Lumpur, Malaysia
[2,3]Advanced Informatics School, Universiti Teknologi Malaysia, Malaysia

***Abstract*—In this paper we investigate the use of multi-process quality model in the adoption of process improvement frameworks. We analyze an improvement effort based on multiple process quality models adoption. At present, there is a possibility of a software development organization to adopt multi-quality and improvement models in order to remain competitive in the IT market place. Various quality models emerge to satisfy different improvement objective such as to improve capability of models, quality management and serve as IT government purpose. The heterogeneity characteristics of the models require further research on dealing with multiple process models at a time. We discuss on the concept of software process and overview on software maintenance and evolution which are important elements in the quality models. The concepts related to process quality model and improvement models are discussed. The research outlined in this paper shows that software processes, maintenance, evolution, quality and improvement have become really important in software engineering. The synergy among the multi-focused process quality model is examined with respect to process improvement. The research outcome is to determine key processes vital to the implementation of multi-process quality model.**

***Index Terms*—evolution; maintenance; multi-process quality model, process improvement**

I. Introduction

IN this paper we investigate the use of multi-process quality model in the adoption of process improvement frameworks. The improvement effort in a multi model environment must be integrated across the quality standards and models which are being used in the software projects or organizations.

Our analysis suggests that due to recent awareness in quality initiatives, there is a possibility of a software development organization to adopt multi-quality and improvement model in order to increase their capability and maturity level. Discussion is first made on the concept of software process and software evolution is presented. We proceed with the concepts related to process quality model and improvement models. The research outlined in this paper shows that software processes, evolution, quality and improvement are primary success factors in software engineering. The synergy among the multi-focused process quality model is examined with respect to process improvement. The research is expected to determine the key processes essential to the implementation of multi-process quality model. Process quality models which are adopted by software development organizations have been proven to improve the software development processes at process, project, and organization levels [3]-[6], [20]-[21]. Organizations also adopt process quality model for achieving quality service [9], [10].

In this paper, the basic concepts related to software processes are firstly discussed. In addition, an overview on software evolution is presented. Furthermore, the basic concepts related to software process quality and improvement models are discussed. The research outlined in this paper indicates the importance of software processes and the use of quality models to manage software processes. This paper presents the discussion on software processes, software maintenance and evolution, software quality and improvement, and process quality models in Section II. Section III discusses about the research method. Section IV describes the research results and discussion, followed by the conclusion in Section V. The paper aims to determine the key processes needed for adoption of multi-process quality model.

II. Related Research

A. *The Concept of Software Processes*

Software processes are defined as a set of interrelated processes in the software development life cycle. A software process establishes a framework for managing software

Manuscript received February 29, 2012. This work was supported by Universiti Lumpur and Universiti Teknologi Malaysia.

A. A. Rahman is with the Software Engineering Section, Malaysian Institute of Information Technology, Universiti Kuala Lumpur, 1016 Jalan Sultan Ismail, 50250 Kuala Lumpur, Malaysia. (Corresponding author: phone: 6-012-6846325; e-mail: aedah@miit.unikl.edu.my).

S. Sahibuddin is with the Advanced Informatics School, Universiti Teknologi Malaysia (e-mail: shamsul@utm.my).

S. Ibrahim is with the Advanced Informatics School, Universiti Teknologi Malaysia (e-mail: suhaimiibrahim@utm.my).

DOI: 10.5176_2010-2283_2.1.152

development activities. The relations between software project and software processes can also be established. Variety of software projects emerged today depending on many domains such as banking, ICT, medical and others. The diversity of software projects require different set of software processes. These software processes produce software development work deliverables and products which comprises of programs, data, and documentation [29]. Most researches have realized the relation between good software products, software processes and practices. This relation is highly required to achieve higher software process maturity levels. The adoption of process quality models plays is a key quality indicator. The Standard for Information Technology – Software Life Cycle Processes in ISO/IEC 12207 Standard defines a software process as a set of interrelated activities transforming inputs to outputs.

B. The Concept of Software Maintenance and Evolution

The need for software change has been tremendously increasing due to rapid growth of software development. Evolution refers to progressive change in software features or attributes. Sommerville [8] defines software evolution as “managing the processes of software system change”. He divided software change and evolution into three strategies which are software maintenance, architectural evolution and software re-engineering. His definition of software maintenance is stated as modifying a program after it has been put into use. According to him, there are three types of software maintenance:

- Corrective - maintenance to repair software faults
- Adaptive - maintenance to adapt software to different operating environment
- Perfective - maintenance to add or modify system’s functionality

Yang and Ward [7] defines software maintenance as:

- Modification of software products to correct faults
- Improve performance or other attributes
- Adapt the product to a changed environment

Other researchers who work on software change and evolution and their related metrics are Lehman and Ramil [23]-[27]. Lehman proposed the Lehman’s Law indicating the laws of software evolution of a software system. The laws include: continuing change, increasing complexity, self regulation, conservation of organizational stability, conservation of familiarity, declining quality and feedback system. Maintenance phase ensures that the system is functioning efficiently and effectively. Maintenance phase is only limited to the original implementation of software life cycle. Thus, only faults during the initial development are fixed. In view of this, we suggest that the “maintenance” is replaced with “evolution”. Through software evolution software practitioners can perceive into a global view of software problems which are highly depending on the rapid changing requirements and environment. Software maintenance phase does not look into this aspect. Bennet and Rajlich [16] in their study proposed five stages which are initial development, evolution, servicing, phase-out and close-down.

In summary, the research works on software maintenance and evolution outlined above has become very vital in software engineering domain. This base of our research is to look into the importance of the software processes and their evolution, which will lead to the area of software process quality, improvement, models and standards.

C. The Concept of Software Process Quality and Improvement

Process quality and improvement may be applied to any discipline. The quality initiatives have emerged as early as 1931 by Shewart including his further work in [32]. Deming [33,34] proposed the cyclic process improvement approach, Plan-Do-Check-Act (PDCA). Basili and Rombach [36] have redefined the PDCA approach specific for the software engineering practices. Each process quality model has its own specific focus and direction such as improving the software products, services, projects or organization. The process quality models are used to assist the organizations to evaluate and improve their software development processes and capabilities. Humphrey [35] initiated early research works on software process improvement which defines the initial step towards improving the capabilities of a software organization. It started by understanding the current status of software development practices. Research on software process improvement via the CMM model has been performed in [20], [21].

D. The Process Quality Models

There are various process quality models emerged today including PDCA, Ishikawa, QIP, IDEAL, Software Capability Maturity Model (SW-CMM), Capability Maturity Model Integration (CMMI), BOOTSTRAP, ISO/IEC 12207, ISO/IEC 15504 (formerly known as SPICE), ISO/IEC 20000, ISO 9000 series and ISO 9126-1. These process quality models are used to manage improvement and process capability of the software development organizations. Among quality experts involved in the quality works are Shewart, Deming, Ishikiwa, Basili, Rombach and Humphrey. The list of process quality models are: IDEAL Model, CMMI, ISO/IEC 20000, BOOTSTRAP, ISO/IEC 15504, ISO/IEC 12207, SW-CMM, Shewart Cycle, Plan-Do-Check-Act (PDCA), Quality Improvement Paradigm (QIP) and Trillium Model.

This paper will focus on reviewing four process quality models to determine and compare relevant maintenance processes suitable for the integrated multi-process quality model. The models are ISO/IEC 12207, ISO/IEC 15504, CMMI, and ISO/IEC 20000.

ISO/IEC 12207 is a reference framework that covers all aspect of the software life cycle processes [15]. The framework covers acquisition of software systems, products and services for the supply, development, operation and maintenance of software products and organization.

ISO/IEC 12207 describes the architecture of the software life cycle o processes. However, the weakness of this framework is, it does not indicate how to implement the activities and tasks in the processes. The framework can be tailored to responsibility of other parties such as the maintenance acquirer and supplier. Five main processes in ISO/IEC 12207 are Acquisition, Supply, Development, Operation, Maintenance and Disposal. Each process consists of set of activities. Each activity contains of set of tasks. ISO/IEC 12207:2008 Systems and Software Engineering – Software Life Cycles consists of 43 systems and software processes. The development, maintenance and operation processes in ISO/IEC 12207 are related to the context of this research which will be discussed in the next section. In ISO/IEC 12207 standard, Improvement process under the Organizational process class is important to assess, measure, control and improve the organizational life-cycle processes. Improvement process contains the following activities: Process Establishment, Process Assessment and Process Improvement.

ISO 15504 process quality model is aligned with the ISO/IEC 12207. In ISO 15504, the activities in Improvement process of ISO/IEC 12207 is described under Process Improvement process group (PIM). ISO 15504 is described by the authors of Capability Maturity Model (CMM), Trillium and BOOTSTRAP process quality models as the most comprehensive software process-oriented reference model [28]. ISO/IEC 15504 is a process improvement and assessment standard originated from a special project called SPICE (Software Process Improvement and Capability dEtermination). ISO/IEC 15504 contains the following life cycle processes: Primary, Organizational and Supporting [14]. ISO 15504 Part 7: “Guide for use in process improvement” has been developed and added to existing process reference model and assessment model. 15504 Part 7 is developed as complementary to the IDEAL improvement model and SW-CMM.

CMMI framework describes discrete levels of process improvement [4]-[6]. CMMI is a capability map which gives description of specific goals and practices that should be attained by an organisation in order to achieve a level of capability and maturity. There are various versions of CMMI now including CMMI v1.2 which consists of three constellations: development (CMMI-DEV v1.2), acquisition (CMMI-ACQ v1.2) and services (CMMI-SVC v1.2). Other works on CMMI have been analyzed in [3], [22], [19]. CMMI has enhanced the assessment approach in BOOTSTRAP and ISO 15504 [28], [14]. CMMI staged representation uses predefined process areas to define improvement path [4], [5]. The CMMI continuous representation allows an organization to perform improvement work for a selected process [5], based on four categories: Process Management (five process areas), Project Management (six process areas), Engineering (six process areas) and Support (five process areas). The table I below presents the processes defined in CMMI staged representation. Further work on interpreting CMMI and its insights have been performed in [19], [22], [30].

TABLE I
PROCESSES IN CMMI – STAGED APPROACH

Level 5 Optimizing	Causal Analysis & Resolution Organizational Innovation & Deployment
Level 4 Quantitatively Managed	Quantitative Project Management
Level 3 Defined	Organizational Process Performance Decision Analysis & Resolution Integrated Project Management Organizational Process Definition Organizational Process Focus Organizational Training Product Integration Requirements Development Risk Management Technical Solution Validation Verification
Level 2 Managed	Configuration Management Measurement & Analysis Project Monitoring & Control Project Planning Process & Product Quality Assurance Requirement Management Supplier Agreement Management
Level 1 Initial	-

Descriptions on some of other process quality models are: Software Capability Maturity Model (SW-CMM) which describes the practices for software process maturity and assessment [21], [22]; BOOTSTRAP that widened the assessment activities scope to the software development organizations and their processes, and practices for evaluating and improving the quality of the organization, project and software development processes. The process areas are organization, methodology and technology [28]; ISO 9001 which describes a model for quality assurance in design, development, production, installing and servicing [11]; ISO 9000-3 that describes guidelines for the application of ISO 9001 in software development [13]; and ISO 9126-1 which describes a model for software product quality consisting of six characteristics such as functionality, reliability, usability efficiency, maintainability and portability [12].

The process quality model for services has been adopted by many organizations such as CMMI–SVC and IT Service Management (ISO/IEC 20000). Hochstein, Tamm and Brenner [2] discussed about issues on services management particularly on the benefit, cost and success factors of service oriented IT management. Improvement in service oriented IT management context is an important factor. ISO/IEC 20000 is the international standard for IT Service Management. The two parts of the standards are: ISO/IEC 20000-1 and ISO/IEC 2000-2 [9], [10]. ISO/IEC 20000 framework can be divided into several processes and their sub-processes as described below:

- Service Delivery Processes – Sub-Processes (Capacity Management, Availability and Continuity Management, Service Level Management, Service Reporting, Information Security Management, Budgeting and Accounting for IT Services)
- Resolution Processes – Sub-Processes (Incident Management, Problem Management)
- Release Processes – Sub-Processes (Release Management)
- Control Processes – Sub-Processes (Configuration Management, Change Management)
- Relationship Processes – Sub-Processes (Business Relationship Management, Supplier Management)

Research on multi-process quality model has been conducted by Rout and Tuffley [31] by assessing two models, the ISO/IEC 12207 and CMMI. Moore [18] also conducted related research on the integration of multi software engineering standards. Cater-Steel, Tan and Toleman [1] in their study also researched on the challenges on adopting multiple process improvement frameworks.

III. RESEARCH METHOD

The research method focuses on the theoretical literature by critically reviewing the multi-process quality models. We perform minor mapping among the existing multi-process quality model. We use the result to determine the most suitable processes for use in the integration of the multi-process quality model. The tables II - III below present the key processes and activities in ISO/IEC 12207:2008 model.

TABLE II
ISO/IEC 12207:2008 – SYSTEMS AND SOFTWARE ENGINEERING – SYSTEM LIFE CYCLE PROCESSES

Agreement Processes	Acquisition Supply
Organizational Project-Enabling Processes	Life Cycle Model Mgt Infrastructure Mgt Project Portfolio Mgt Human Resource Mgt Quality Mgt
Project Processes	Project Planning Project Assessment and Control Decision Mgt Risk Mgt Configuration Mgt Information Mgt Measurement
Technical Processes	Stakeholder Requirements Definition System Requirements Analysis System Architectural Design Implementation System Integration System Qualification Testing Software Installation Software Acceptance Support Software Operation Software Maintenance Software Disposal

TABLE III
ISO/IEC 12207: 2008 – SYSTEMS AND SOFTWARE ENGINEERING – SOFTWARE LIFE CYCLE PROCESSES

Software Implementation Processes	Software Implementation Software Requirements Analysis Software Architectural Design Software Detailed Design Software Construction Software Integration Software Qualification Testing
Software Support Processes	Software Documentation Management Software Configuration Management Software Quality Assurance Software Verification Software Validation Software Review Software Audit Software Problem Resolution
Software Reuse Processes	Domain Engineering Reuse Asset Management Reuse Program Management

The table IV below presents the key processes and activities in ISO/IEC 15504-5:2006 process quality models.

TABLE IV
PROCESSES CATEGORIES &GROUPS IN ISO/IEC 15504-5: 2006 (A)

PRIMARY LIFE CYCLE PROCESSES	
Acquisition Process Group	Acquisiton Preparation Supplier Selection
Supply Process Groups	Contract Agreement Supplier Monitoring Customer Acceptance Supplier Tendering
Engineering Process Group	Requirements Elicitation System Requirements Analysis System Architectural Design Software Requirements Analysis Software Design Software Construction Software Integration Software Testing System Integration System Testing Software Installation Software and System Maintenance
Operation Process Group	Operational Use Customer Support
ORGANIZATIONAL LIFE CYCLE PROCESSES	
Management Process Group	Organizational Alignment Organizational Management Project Management Quality Management Risk Management Measurement
Process Improvement Process Group	Process Establishment Process Assessment Process Improvement
Resource and Infrastructure Process Group	Human Resource Management Training Knowledge Management Infrastructure
Reuse Process Group	Asset Management Reuse Program Management Domain Engineering

TABLE IV
PROCESSES CATEGORIES & GROUPS IN ISO/IEC 15504-5: 2006 (B)

SUPPORTING LIFE CYCLE PROCESSES	
Support Process Group	Quality Assurance Verification Validation Joint Review Audit Product Evaluation Documentation Configuration Management Problem Resolution Management Change Request Management

The table V below presents the structure comparison of the selected process quality models in ISO/IEC 15504, CMMI-DEV and ISO/IEC 20000 process quality models.

TABLE V
COMPARISON OF STRUCTURE OF THREE SELECTED PROCESS QUALITY MODELS

PROCESS QUALITY MODELS AND FOCUS	PROCESSES/LEVELS
ISO/IEC 15504 (System and Software)	Primary Life Cycle Processes, Organizational Life Cycle Processes, Supporting Life Cycle Processes
CMMI-DEV (Software Development)	Levels in Staged approach – Initial, Repeatable, Defined, Managed and Optimizing Process Categories in Continuous approach – Process, Project, Engineering, Support
ISO/IEC 20000 (Services)	Service Delivery Processes, Resolution Processes, Release Processes, Control Processes Relationship Processes

IV. RESULTS AND DISCUSSION

The results from the mapping made on the process quality models are used to determine the processes for use in the integration of multi-process quality models. There synergy among the models have derive several processes that need to be considered for the adoption of multi-process quality model. We determined from our research that CMMI-DEV v1.2 focuses on improvement of development processes only, and not on post-deployment processes. Configuration Management is identified as a process that can be linked to maintenance process. ISO/IEC 12207 and ISO/IEC 15504 have included the maintenance, support and process improvement processes. As these are all from the viewpoint of development, we are also looking into the view of services using ISO/IEC 20000. ISO/IEC 20000 provides a holistic view of service operation and management although lack improvement path. The processes that we derive from the mapping are development, maintenance, and support processes.

V. CONCLUSION

This paper has reviewed recent studies in multi-process quality models to manage software process improvement of the software development organizations. They aim to leverage the benefits of each model to develop a common solution for their quality practice. The adoption of multi-process quality models bring new challenges that need to be handled by the software development organizations. For this reason, we have conducted comparison among four process quality models and discover several issues related to process improvement for development and operation. Despite the commonalities obtained, we encounter significant differences in terminologies, structures, purpose and direction among the models. Relevant key processes for multi-process quality models have been identified and extracted to support the integration approach.

ACKNOWLEDGMENT

The authors wish to thank Universiti Kuala Lumpur and Universiti Teknologi Malaysia for the research support.

REFERENCES

[1] A. Cater-Steel, W.G. Tan, and M. Toleman, “Challenge of Adopting Multiple Process Improvement Frameworks”, In: 14th European Conference on Information Systems (ECIS2006), Goteborg, Sweden, 12-14 June 2006.

[2] A. Hochstein, G. Tamm, and W. Brenner, “Service oriented IT management: Benefit, cost and success factors”, European Conference on Information Systems (ECIS 2005), Proceedings 13, Regensburg, Germany, 2005.

[3] B. McFeeley, “IDEAL^{SM}: A user’s guide for software process improvement”, CMU/SEI-96-HB-001, Pittsburg, Pennsylvania

[4] CMMI Product Team, “Capability Maturity Model Integration, Version 1.1, CMMI-SW/SE/IPPD/SS, Staged Representation”, Technical Note, CMU/SEI-2002-TR-011, 2002.

[5] CMMI Product Team, “Capability Maturity Model Integration, Version 1.1, CMMI-SW/SE/IPPD/SS, Continuous Representation”, Technical Note, CMU/SEI-2002-TR-012, 2002.

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

[7] H. Yang and M. Ward, “Successful evolution of software system”, London: Artech House, 2003.

[8] I. Sommerville, Software Engineering, 9th ed., England: Pearson Education Limited, 2010.

[9] International Standard Organisation, “ISO/IEC 20000-1. Information Technology – Service Management – Part 1: Specification”, International Standard Organisation, Geneva, Switzerland, 2005.

[10] International Standard Organisation, “ISO/IEC 20000-2. Information Technology – Service Management – Part 2: Code of Practice”, International Standard Organisation, Geneva, Switzerland, 2005.

[11] ISO 9001, “Quality management system – Fundamentals and vocabulary”, 2000.

[12] ISO/IEC 9126-1, “Software Engineering – Product Quality – Part 1 : Quality Model”, 2001.

[13] ISO 9000-3, “Guidelines for the application of ISO 9001:1994 to the development, supply, installation and maintenance of computer software”, 1997.

[14] ISO/IEC 15504-6, “Information Technology – Process Assessment – An exemplar process assessment model”, 2006.

[15] ISO/IEC 12207, “Information Technology – Software life cycle processes”, Geneva, Switzerland, 2002.

[16] K.H. Bennet and V.T. Rajlich, “Software Maintenance and Evolution: A Roadmap”, In: International Conference of Software Engineering (ICSE2000): Proceedings of the Conference of the Future of Software Engineering, New York, NY, USA: ACM Press, pp. 73-87, 2000.

[17] K. Ishikawa, What is Total Quality Control? The Japanese Way, Eaglewood Cliffs, N.J: Prentice-Hall, 1985.

[18] J. Moore, “An integrated collection of software engineering standards”,IEEE Software, vol. 16(6), pp. 51-57, 1999.

[19] L.G. Jones and A.L. Soule, “Software process improvement and product line practice: CMMI and the framework for software product line practice”, Technical Note, CMU/SEI-2002-TN-012, 2002.

[20] M.C. Paulk, C. Weber, B. Curtis, and M.B. Chrissis, “Capability Maturity Model for Software, Version 1.1”, Technical Report, Software Engineering Institute, Carnegie Mellon University, CMU/SEI-93-TR-024 ESC-TR-93-177, 1993.

[21] M.C. Paulk, C. Weber, B. Curtis, and M.B. Chrissis, “The Capability Maturity Model: Guidelines for improving the software process, Version 1.1”, Boston: Addison Wesley, 1995.

[22] M.K. Kulpa and K.A. Johnson, Interpreting the CMMI: A Process Improvement Approach, Auerbach Publications, 2003.

[23] M.M. Lehman, “Laws of software evolution revisited”, Lecture Notes in Computer Science 1149, Berlin: Springer, pp.108 – 124, 1997.

[24] M.M. Lehman, D.E. Perry, and P.D. Wernick, “Metrics and Laws of the Software Evolution”, Proceedings of the 4th International Software Metrics Symposium, IEEE, 1997.

[25] M.M. Lehman and J.F. Ramil, “The Impact of Feedback in the Global Software Process”, The Journal of Systems and Software, vol. 46, pp. 123-134, 1999.

[26] M.M. Lehman and J.F. Ramil, “Towards a theory of software evolution and its practical impact”, Proceedings of International Symposium on the Principles of Software Evolution, IEEE Computer Society, Washington, pp. 2-11, 2000.

[27] M.M. Lehman and J.F. Ramil, “Behavioural modelling of lon-lived evolution processes: Some issues and an example”, Journal of Software Maintenance: Research and Practice 14, pp. 335 – 351, 2002.

[28] P. Kuvaja, J. Simila, A. Krzanik, S. Saukkonen, and G. Kochi, Software Process Assessment and Improvement – The BOOTSTRAP Approach, Oxford, UK: Blackwell Publisher, 1994.

[29] R. Pressman, Software Engineering: A Practitioner’s Approach, 7th ed., New York: McGraw-Hill International, 2009.

[30] T. Kasse, Practical Insight into CMMI, Norwood, MA: Artech House, 2004.

[31] T.P. Rout and A. Tuffley, “A harmonizing ISO/IEC 15504 and CMMI”, Software Process Improvement and Practice, vol. 12(4), pp. 361-371, 2007.

[32] W.A. Shewart, “Statistical method from the viewpoint of quality control”, The Graduate School, George Washington University, Washington D.C., 1939.

[33] W.E. Deming, Elementary principles of the statistical control of quality. JUSE, 1950.

[34] W.E. Deming, “Out of crisis: Quality, productivity and competitive position”, MIT Center for Advanced Engineering Study, Cambridge, MA, 1986.

[35] W.S. Humphrey, Managing the Software Process. Reading, Massachusetts: Addison-Wesley, 1989.

[36] V.R. Basili and H.D Rombach, “The TAME project: Towards improvement oriented software environments”, IEEE Transactions on Software Engineering. SE-14(6):758773, 1988.

Aedah Abd Rahman This author was born in Kuala Lumpur, Malaysia in 1974. Aedah obtained her Masters and Bachelor degrees from University of Malaya, Kuala Lumpur, Malaysia.

Masters of Computer Science (Specialization: software engineering), University of Malaya, Kuala Lumpur, 2002

Bachelor of Computer Science, University of Malaya, Kuala Lumpur, 2002

She is currently a Lecturer in Software Engineering Section, Malaysian Institute of Information Technology (MIIT) at Universiti Kuala Lumpur, Malaysia. Over the past twelve years she has been lecturing at Universiti Tun Abdul Razak (UNITAR), Petaling Jaya, Malaysia and Open University Malaysia, (OUM) Kuala Lumpur, Malaysia. Her works and experiences in software engineering field include as an academician, researcher, Head of Software Engineering Department at UNITAR and Program Coordinator for Software Engineering Masters and Bachelors Program at OUM. She has published various publications during the past twelve years. The recent lists of published articles are:

A.A. Rahman, S. Sahibuddin & S. Ibrahim (2011), "A Study of Process Improvement Best Practice", IEEE International Conference of Information Technology and Multimedia 2011 (ICIMU2011), Universiti Tenaga Nasional (UNITEN), 14 - 16 November 2011.

A.A. Rahman, S. Sahibuddin & S. Ibrahim (2011), "A Multi-Process Quality Model", Proceedings of the 2nd Annual International Conference on Software Engineering and Applications 2011 (SEA2011), Fort Canning Hotel, Singapore, 12 - 13 December 2011. ISSN: 2251-2217

A.A. Rahman, S. Sahibuddin & S. Ibrahim (2011), "A Unified Framework for Software Engineering Process Improvement - A Taxonomy Comparative Analysis", IEEE Malaysian Software Engineering Conference 2011 (MySEC2011), Puteri Pacific Hotel, Johor, 13 - 14 December 2011. IEEE/ISBN: 978-967-0194-16-5

A.A. Rahman, S. Sahibuddin & S. Ibrahim (2011), “Examining Software Process Quality Management Frameworks and Their Relevance Towards Software Evolution”, International Conference of Computer Engineering and Technology 2010, Jodhpur, Rajasthan, India, 13-14 November 2010.

Her current research area is Software Process Quality and Improvement. Other research areas include Software Quality and Process Evolution and Software Requirements.

Ms Aedah is a member of Malaysian Software Engineering Interest Group (MySEIG).

Shamsul Sahibuddin This author is currently a Professor at Universiti Teknologi Malaysia (UTM). Prof. Dr. Shamsul is the Dean of Advanced Informatics School at Universiti Teknologi Malaysia, Malaysia.

Suhaimi Ibrahim This author is currently an Associate Professor in Advanced Informatics School at Universiti Teknologi Malaysia, Malaysia.