A COMPARATIVE STUDY OF SOFTWARE ENGINEERING PROCESS MODELS FOR MIDDLE EAST AIRLINES

Hasan Sahily and Ramzi A. Haraty
Lebanese American University
P.O. Box 13-5053 Chouran
Beirut, Lebanon 1120 2801
Email: rharaty@lau.edu.lb

ABSTRACT
The aim of this paper is to investigate the theoretical and managerial foundation of software engineering process models mainly CMM, SPICE, ISO 9001, BOOTSTRAP, and SEPRM; assess their applicability; evaluate each one, and make recommendations about the best process to be followed and applied in Middle East Airlines. We also aim to explore the feature, orientation, interrelationship, and transformability of existing process models, and evaluate the integration between the existing process models and methodologies into a unified process model.

KEY WORDS
BOOTSTRAP, CMM, ISO 9001, SEPRM and SPICE.

1. Introduction

Software engineering is a discipline of increasing importance in computing. The main problems in software engineering occur as a result of complexity to set and determine business requirements, generalization of software products, and the heavily interactive dependency of software, hardware, and human beings.

A new approach for dealing with the difficulties of large-scale software development emerged in the last two decades. It required establishing appropriate software engineering process system that is based on a group of best practices in software development, institutions, and management, which serves as a reference model for regulating the process activities in software development companies.

To model the software engineering processes, a number of software process models such as ISO 9001, CMM, SPICE, SEPRM and BOOTSTRAP have been developed in the last 20 years [2][3][9][10][15]. Studies in the software process recommend that a move is needed from control of the quality of the final software product to optimizing the processes that produce the software. It is also well-known that the software engineering process can be well-standardized, established, stabilized, and reused.

The remainder of this paper is organized as follows: Section 2 describes the fundamentals of the software engineering process. Section 3 describes the standards. Section 4 provides a comparison of the different models. Section five presents two cases studies with Middle East airlines. And section 6 provides a conclusion.

2. Fundamentals of the Software Engineering Process

Software engineering adopts engineering approaches to develop large-scale software with high efficiency, quality, reliability, less cost, and measurable time frame of software development.

Currently, it is necessary to simplify the existing process models and create a unified structure and introduce formal and algorithmic description of an integrated process. Organizing and implementing software engineering through software processes is a strategically important approach in the software industry. The objective of this work is to discover the different software engineering process domain and its architecture. An assessment of a unified software engineering process systems model will be applied, focusing on the weaknesses and strengths in current process models. Main areas in process analysis, design, development, implementation, deployment, assessment and improvement may not have been covered by existing process models.

3. Software Process Standards

In this section, we explain the assessment and improvement methodology and then go into details of standard processes that will cover the following process assessment methods ISO 9001, CMM, SPICE, BOOTSTRAP, and the hybrid model SEPRM.

3.1 Assessments and Improvement

Software engineering and software process improvement standards are important as they are recognized by the software developers because they transfer best practice into industry. Companies acquiring software focus on standards because they ask for level of quality that must be respected during the development life cycle. Those standards are considered necessary for organization and product to be certified.

3.2 Standard Processes

3.2.1 ISO 9000-1

ISO 9001-1 provides a set of standards to manage the production tasks. Organizations have to set a quality control to guide all phases of production and delivery process. These include:

- Audit of project to comply by quality control.
- Improve the system quality.
- Provide new thoughts to developers to set new standards and procedures to improve the overall quality of the software.

A quality manual contains the details of the quality system, such as the development activities. The project manager defines the quality issues related to the project such as procedures, plan, and resources. Note that ISO-9001 is specialized in software production [5]. The reason behind that is the following:

- Quality control has to be applied in purchasing, development, and maintenance phases.
- The company that wants to buy software must coordinate with the software supplier.
- The supplier has to define the quality control and ensure that the whole organization understands and implements the software.

ISO 9000-1 does not require a pre-determined steps and evaluation method of quality; it might be supported by spiral development methodology [6]. ISO 9000-1 might be used during contractual phase between a software developer house and a client to specify quality elements as part of the supplier's quality system, where supplier commits to apply the quality principles that already stated in the contract. Also, the supplier states the quality standards to compete with other supplier software. ISO models a software process system in three process subsystems, 20 main topics areas (MTA), and 177 management issues (MI). ISO is a parallel process model with three subsystems at the system level. Each subsystem can be extended to a number of parallel processes (MTAs). Each MTA can be extended to a number of MIs in a similar way [11].

Assessment is done through verification of all MIs according to the rating scale that is either satisfied (1) or non-satisfied (0). The total satisfied number of Quality System Attributes (QSAs) is counted at every level and at all levels. The capability level of the project is considered "Pass" if it passes the all 177 MIs.

3.2.2 The Capability Maturity Model

During 1989, the Software Engineering Institute (SEI) initiated the software process assessment project [4]. It emphasized the idea to consolidate and improve processes through assessment. Two programs were in place:

- **Software-process Assessment Program**. Used by organizations that need to evaluate their process for improvement. The output provides the organization with recommendations on how to carry out process improvement.
- **Software Capability Evaluation Program**. Practiced by big clients such as departments of US government that evaluates the level of maturity of the suppliers. A grade between one and five is determined that specifies the maturity level of the organization. The assessment process is done by an external company.

To make assessment of organization's maturity, SEI sets five levels of software maturity and defined a set of characteristics for each level. To reach the next level, a certain goal has to be achieved:

- Level 1: success rely on the developers skills.
- Level 2: processes are repeatable. A company has to set the project management policies and procedures to follow throughout the project life cycle. A quality assurance function monitors the practice of policies and procedures. It ensures repeating previous success on similar projects.
- Level 3: defined level - identify the processes related to the software engineering and project management, which are customized to meet the needs of each project.
- Level 4: the managed level - it measures the quality of product and process in a quantifiable way. This way, the project manager is able to determine the reasons of exceptional events and rectify them.
- Level 5: optimizing level - continuous improvement of processes based on results of previous instants. It is done when introducing new technologies and methods.

The first step of Process Assessment Program is to train the team who performs the assessment and then this team will choose members of the project who will fill the SEI questionnaire, and the assessment team will interview the project's members and make assessment and define the weaknesses of the company. Top management will see the results of the assessment. The main weaknesses of the Software Capability Evaluation program is that a software development organization will not move to higher level even if it fulfils the needed requirements for that level, but does not fulfil the requirements of the lower level.

CMM models a software process system at five capability levels, in 18 key practice areas, and 150 key practices (KP). KP performance rating is according to the table 1:

Table 1. KP performance ratings

Scale	Description	Rating entry
4	Yes	>=80%
3	No	< 80%
2	Doesn't apply	-
1	Don't know	-

Rate performance of KP according to the practice performance scale, if rating >=80 or KP doesn't apply then KP is satisfied and we add 1 to satisfied cumulative level. Process capability determination is done according to each level; it has to satisfy at least 80% of a previous level to proceed to the next level.

3.2.3 BOOTSTRAP

The European Union supported BOOTSRAP project that initiated in 1989 and had an objective to introduce software technology into industry. The project aimed to define methodology to assess European industries in different domains, such as insurance, banking, administration [12]. The approach was that modern technology is not effective unless it is supported with methods to build solutions and must be implemented within a strictly organized process. Initially, BOOTSRAP took some concepts from the ISO 9000-1 and CMM (quality management and maturity level). BOOTSTRAP offers absent-weak, basic-present, significant-fair, and extensive-complete ratings.

BOOTSTRAP questionnaire is built on well-organized attributes. In fact, it is based on the same five levels of maturity; whereas it evaluates a maturity level for each quality attribute, so that companies can define the problems and rectify them. Bootstrap assessment is also used to evaluate if the organization is ready to be certified by ISO 9000. Organizations need to show that they have some methodologies and use them and accordingly, certification is given to a company that is between levels two and three [8].

From the functional approach, BOOTSTRAP represents a software process system in three process areas (Pas), nine process categories (PCs), 32 processes (PRs), and 201 QSAs. It is a parallel process model at the level of the system where each process area is divided to a number of parallel PCs. Each PC is divided to a number of QSAs. Table 2 shows the number of defined QSA and accumulated pass thresholds at each level.

Table 2. QSA and pass thresholds

Level	N_{QSA} number of defined QSA	P'_{QSA} cumulated pass thresholds at each level
1	0	0
2	40	32
3	81	97
4	27	119
5	53	162

QSA performance rating is done according to the table 3:

Table 3. QSA performance ratings.

Scale	Description	Rating entry
4	Complete / extensive	>=80%
3	Largely satisfied	between 66.7% and 79.9%
2	Partially satisfied	Between 33.3% and 66.6%
1	Absent / Poor	<=33.2%
0	Doesn't apply	-

QSA is satisfied if the rate is 3, 4, or doesn't apply. Process capability determination is measured by counting total satisfied number of QSAs (Nsat) at all levels compared to the cumulated pass thresholds at each level

- If ($N_{SAT} < P'_{QSA}[2]$), then project at Initial level
- if ($N_{SAT} < P'_{QSA}[3]$), then project at Repeatable level
- if ($N_{SAT} < P'_{QSA}[4]$), then project at Defined level
- if ($N_{SAT} < P'_{QSA}[5]$), then project at Managed level
- else project at Optimized level .

3.2.4. SPICE

SPICE (Software Process Improvement and Capability determination) is funded by International Committee on Software Engineering Standards ISO/IEC JTC 1/SC7 [13]. Its main objective is building a standard for software process assessment that covers development, management, purchasing, quality, customer support, technology transfer, and human concerns. It uses assessment methods, such as CMM, BOOTSTRAP, and ISO 9000-1. The output of the project is the new ISO 15504 SPICE standard [1]. It helps in defining:

- High level of goals and activities that characterize a successful software engineering with high capabilities.
- Training of the evaluator and set the procedures for evaluation.
- Process assessment and enhancement stages.
- Determine the capabilities of a company after the assessment process.
- Determine the business risks for new software or product.

ISO/IEC TR 15504 models a software process system in 5 process categories, 35 processes, and 201 base practices. The performance rating scale is shown in table 4:

Table 4. Performance rating scale for SPICE

Scale	Description	Rating entry
4 (F)	Fully achieved	between 86% and 100%
3 (L)	Largely achieved	between 51% and 85%
2 (P)	Partially achieved	Between 16% and 50%
1 (N)	Not achieved	Between 0% and 15%

To compute the project capability determination, we have to find the cumulative rate average, and the final result as shown in table 5.

Table 5. Cumulative rate average and result for SPICE

Level	Key challenges	Result
5 Optimizing	Still human intensive process Maintain Organization at Optimizing level	**Productivity** **&** **Quality**
4 Predictable Managed	Changing Technology Problem Analysis Problem Prevention	
3 Defined Established	Process Measurement Process Analysis Quantitative Quality Plans	
2 Repeatable	Training, Testing Technical Practices & Reviews Process focus, Standards & Processes	
1 Initial	Project Management & Planning Configuration Management Software Quality Assurance	**Risk**

3.2.5 Software Engineering Process Reference Model (SEPRM)

The main objective of SEPRM is offering a mature and integrated software engineering process reference model. SEPRM is a complete 2-D software engineering process system model that integrates the advantages of the existing process models [4]. Its process capability model is independently operational with a unique process capability scale. The current process models emphasis conflicting areas of the process domain. Some of its vital parts are not covered by ISO, BOOTSTRAP or SPICE.

SEPRM provides the means to integrate and unify the current process models, such as ISO 9001, CMM, BOOTSTRAP, and SPICE, by a well-founded process structure, a standard superset of base process activities (BPAs), and a stable transformable capability determination algorithm [7]. SEPRM provides a comprehensive process reference model. It develops a process capability determination methodology that is relatively lower in operating complexity and easier for application in process assessment and improvement. It allows software development companies to associate their capabilities to others using different process models.

SEPRM organizes the processes into three process-subsystems, 12 process-categories, 51 processes, and 444 Base-Process-Activities "BPA". In the SEPRM process model, each process subsystem can be divided downwards to a number of PCAs, then to a number of PROCs. Furthermore, each PROC can be divided to a number of BPAs in a similar way. BPAs practice performance ratings are depicted in table 6:

Table 6. Performance rating for SEPRM

Scale	Description	Rating entry
(F) **5**	Fully adequate	**%100 - %90**
(L) **3**	Largely adequate	**%89 - %60**
(P) 1	Partially adequate	25% - 59%
(N) 5	Not adequate	0% - 24%

4. Comparison between the Different Processes

To make a comparison between the different process models, it is easier to use an intermediary reference model to simplify the many-to-many mapping into a many-to-one to reduce the difficulty of shared map of different models. To have a more precise mapping, it has to be done at the BPA level. The terminology of BPA differs from model to model, such as:

- CMM→ key practice
- ISO 9001→ management issue
- BOOTSTRAP→ quality system attribute
- ISO/IEC 15504 → base practice

Those process elements are referred as base process activities in SEPRM. When we use an intermediary reference model as shown in the figure 1, it will reduce the difficulty of mapping between the models.

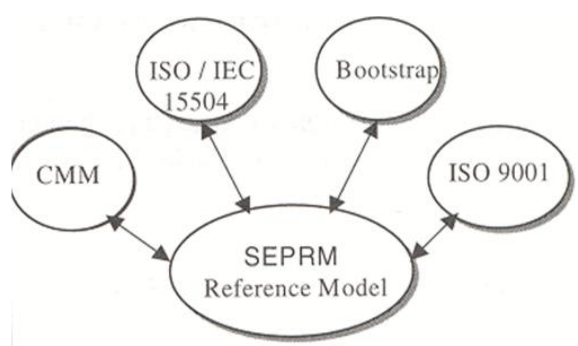

Figure 1. The role of a software process reference.

The power and completeness of a process model is defined by both its process domain and its capability determination methodology. The process domain defined in SEPRM consists of 444 BPA. The BPAs are equivalent to 177 management issues (MIs) in ISO 9001, and 150 key practices (KPs) in CMM and 201 quality system attributes (QSAs) in BOOTSTRAP and 201 base practices (BPs) in SPICE.

Compatibility between a numbers of process models is defined as the degree of joint domain coverage, which is determined by the sets of BPAs of the process models. The compatibility-adaptability degree, C_k, can be described at five levels as follows:

- C1: specific BPAs identified in one model.
- C2: Shared BPAs in 2 models.
- C3: Shared BPAs in 3 models.
- C4: Shared BPAs in 4 models.
- C5: BPAs shared in 5 models.

There are 729 equivalent BPAs independently known in ISO/IEC TR 15504, CMM, BOOTSTRAP, and ISO 9001. When we do a filter of overlaps and redundant among them, we will have 407 independent BPAs extracted out of the 4 models. Those BPAs are the major set of SEPRM model.

In the SEPRM process domain, there are 37 new BPAs (8%) that are defined in the SEPRM model, that focus on evaluation of software development methodology, use new tools to facilitate management, development, reusability, test, maintenance, and documentation.

The compatibility of SEPRM to ISO/IEC TR 15504, CMM, BOOTSTRAP, and ISO 9001 is derived as shown in table 7.

Table 7. Compatibility Degree of SEPRM to other Process Models

	Subsystem - Organization			
Compatibility	1	2	3	C_k(SEPRM)
C_1(SEPRM)	1	17	19	37
C_2(SEPRM)	56	70	143	269
C_3(SEPRM)	20	22	58	100
C_4(SEPRM)	2	6	24	32
C_5(SEPRM)	2	0	4	6

Compatibility between SEPRM with other models is at five levels:

- Level 1: 37 BPAs new defined in SEPRM model, and focus on development and management of subsystems.
- Level 2: 269 BPAs (approximately half of the process domain) are only found in SEPRM and one of the other models; it also means that without SEPRM the compatibility between the other 4 models could be quite low.
- Level 3: 100 BPAs are found in SEPRM and 2 other models.
- Level 4: 32 BPAs are found in SEPRM and 3 other models.
- Level 5: 6 BPAs are found with the highest compatibility.

System level relationship between SEPRM and ISO/IEC TR 15504, CMM, BOOTSTRAP, and ISO 9001 are listed in table 8, where the symbol "X" represents one of the other four models where appropriate. SEPRM is 100% interconnected to all ISO/IEC TR 5504, CMM, BOOTSTRAP, and ISO 9001, but not vice versa. The BPA is the fundamental element in modelling a software process. By observing the configurations of BPAs in current process models, orientation and emphasis of these models can be explored quantitatively and objectively. The different patterns of configurations of the BPAs show different orientations and concentrate on current process models. It is obvious that SEPRM provides enhanced BPAs in all of the three process subsystems.

Table 8. SEPRM vs. ISO/IEC TR 15504(SPICE), CMM, BOOTSTRAP, and ISO 9001

Relation ship	***Subsystem 1 Organization***	***Subsystem 2 Development***	***Subsystem 3 Management***	***r(SEPRM, X)***	***ρ(SEPRM, X)***
SPICE	*61*	*47*	*93*	*201*	*100%*
CMM	*21*	*20*	*109*	*150*	*100%*
BOOTSTRAP	*23*	*64*	*114*	*201*	*100%*
ISO 9001	*25*	*44*	*108*	*177*	*100%*

Relative BPA configurations of current process models in the organization, development, and management process subsystems can be derived, according to the distributions of percentages of BPAs within individual models.

We note the following:

- CMM is a management-oriented process model.
- BOOTSTRAP is a technical-oriented process model.
- ISO/IEC TR 15504 is an organization-oriented process model.

To implement process-based software engineering in large-scale software development, the deployment of effort and resources in organization, development, and management might be considered as 20% - 30% - 50%. It is important to mention that traditional software development has been concentrated only on the technical processes, whereas large-scale software development requires extensive resources and efforts has to concentrate on organization infrastructures, management measures, and software quality assurance in software engineering [14].

5. Case Studies

5.1 Case study 1 - Frequent Flyer Program

The Frequent Flyer system allows MEA to provide passengers with greater recognition and facilities when they use MEA flights and its partners. It also, rewards passengers by keeping track of ticket purchasing and utilization and provide more benefits and services to its loyal passengers. This program provides passengers an extra point for every mile flown. Passengers get a free reward ticket when they accumulate a certain level of points or a predefined number of flights. MEA also offer its loyal passengers many privileges such as upgrade, access to airport lounge and special check-in counters.

This system is developed with the help of Air France. The Oracle database is hosted in Valbone (France). Communication takes place worldwide through the Sita network. MEA developed part of the project that concerns the customer relationship:

- Sending the kits.
- Sending the cards.
- Calculation of earned miles.
- Interface with the database, revenue accounting, financial accounting, departure control system, etc.

Development resources

- Development time: 11 months.
- Resources: 1 designer, 2 programmers.
- Database: Oracle and Jet database.
- Programming language: Visual Basic.
- Number of tables: 96.
- Number of reports: 56.

Results

Implementing the different process model to test the effectiveness of the system showed that CMM, and ISO do not provide accurate result compared with the hybrid solution SEPRM. On the other hand, the result of the BOOTSTRAP is more closely to the hybrid SEPRM.

1. The work evaluated using SEPRM has a result of 2.67, which is in the higher level of the partially adequate or satisfied, whereas in CMM showed that it’s at the initial level.

- The reasons are that it did not pass the repeatable level (it needs 11 points) although it satisfies around 90% of the remaining levels (Defined, Managed, and Optimized).
- CMM does not include a large factor of assessing criteria such as the SEPRM (150 against 444).

2. ISO 9001 does not provide a clear idea about the assessment of the project. It evaluates the project either by fail or pass. To be "Passed", it needs to satisfy all the key factors. There is no intermediate solution.

- SEPRM evaluated the frequent flyer application to be on the higher level of the partially adequate or satisfied, whereas in ISO9001 evaluated as "Failed" because it did not satisfy the 14 points of 177 (passing factors were 163 out of 177).
- ISO does not include a large factor of assessing criteria such as the SEPRM (177 against 444).

3. BOOTSTRAP evaluated the project to be in the repeatable level, which is similar to the SEPRM. Satisfied factors were 101 out of 201.

- The reasons are: it includes more factors than ISO 9001 and CMM, and therefore the evaluation criteria are more close to reality (201 compared to 444).
- Its disadvantage is that, it does not include all the factors such as the SEPRM.

5.1 Case study 2 - Rostima (Roster System)

Rostima enterprise plus has been designed by Rostima to automate and optimise the time-consuming task of scheduling. It is an adequate solution for planning and scheduling of employees attendance and distribution among different tasks. The problem with this system is that it was not fair in distributing shifts among employees. Moreover, it is that adequate in covering all the activities in the airport. MEA wanted to improve it either by buying other modules from Rostima system provider, which will cost MEA lots of money, or by depending on its own resources to improve the system. MEA went in to the next choice by developing a new system called Rostimo, which run in parallel with Rostima, by which its main purpose is as follows:

- Do a fair distribution of shifts among staff, after exporting the schedule from Rostima.
- Covering all the activities in a sufficient way.
- Taking into consideration staff constraints.
- Ability to manage resources in a better way.
- Interface with the database and activity distribution.

Development resources

- Development time: 8 months.
- Resources: 2 programmers.
- Database: Access database
- Programming language: Visual Basic.net
- Number of tables: 20
- Number of reports: 30

Results

Implementing the different process model to test the effectiveness of the system showed that CMM, and ISO do not provide accurate result compared with the hybrid solution SEPRM. The result showed that BOOTSTRAP is close to the hybrid SEPRM.

1. The project evaluated using SEPRM has a result of 2.98, which is in the higher level of the partially adequate or satisfied, whereas in CMM showed that it is at the initial level.

- The reasons are that it did not pass the repeatable level (it needs 16 points) although it satisfies around 81% of the remaining levels (Defined, Managed, and Optimized).
- CMM does not include a large factor of assessing criteria such as the SEPRM (150 against 444).

2. ISO 9001 does not provide a clear idea about the assessment of the project. It evaluates the project either by fail or pass. To be "Passed", it needs to satisfy all the key factors. There is no intermediate solution.

- SEPRM evaluated the ROSTIMA application to be on the higher level of the partially adequate or satisfied, whereas in ISO9001 evaluated as "Failed" because it did not satisfy the 27 points of 177 (passing factors were 150 out of 177).
- ISO does not include a large factor of assessing criteria such as the SEPRM (177 against 444).

3. BOOTSTRAP is the process model that mostly approximate to the SEPRM, it evaluated the project to be in the repeatable level, which is similar to the SEPRM. Satisfied factors were 103 out of 201.

- It includes more factors than ISO9001 and CMM, and therefore the evaluation criteria are more close to the reality (201 compared to 444).
- Its disadvantage is that, it does not include all the factors such as the SEPRM.

For more information on case studies 1 and 2, interested readers are referred to [16].

6. Conclusion

The software process was recognized as a valid element of software engineering only a few years ago. This project has attempted to show that the software engineering process as a system is an ideal means and a powerful tool for the infrastructure of software engineering. The software industry has grown rapidly to become one of the most important labour-intensive industries. It is quite encouraging to see that more and more software development organizations have adopted software process systems as the key architectural structures for implementing and improving software engineering. In the future, we can expect more software development organizations and projects operating at higher capability levels in process-based software engineering.

We conclude that the current methodologies that cover individual sub-domains are insufficient. So, a unified approach has been required to accommodate the full range of practices and requirements. The two case studies that we implemented in MEA and integrate most of the existing software processes shows with no doubt that a unified process model is ideal to enhance software system and best assessment approaches to software engineering. Many concepts and theories in software engineering such as specification elicitation, requirement analysis, design, process, testing, and quality assurance were derived by methods practiced in other engineering domains. Therefore, comparative studies between different areas from wide views of the engineering domains could give new ideas and better understanding of how software engineering differs from other engineering disciplines, and how software engineering could gain experience and benefits from other disciplines.

Acknowledgement

This work was funded by the Lebanese American University.

References

[1] K. El-Eman and Ho-Won Jung. An Empirical Evaluation of the ISO/IEC 15504 Assessment Model. *Journal of Systems and Software*. Volume 59, Issue 1, 15 October 2001, 23-41.

[2] Y. Wang & G. King. *A new Approach to Benchmark Based Process Improvement. Rated Newspaper*. Supported by EU Project. http://www.iscn.at/select_newspaper/assessments/southampton.html

[3] W. Emmerich, *Software Process: Principles, Methodology, and Technology* (Publisher Springer US Heidelberg USA,1999).

[4] *Describing the Capability Maturity Model*. Gartner Measurement. http:/www.gartner.com/measurement.

[5] M. Paulk, *How ISO 9001 Compares with the CMM*. Article sponsored by the US Department of Defence under Contract F19628-90-C-003. http://home.comcast.net/~mark.paulk/papers/p1995a.pdf,1995.

[6] R. Bamford & W. Deibler. *Hybrid Multi-Model Assessment. When the CMM Meets ISO 9001*. Crosstalk - the Journal of Defence Software Engineering. http://www.docstoc.com/docs/420649/Hybrid-Multi-Model-Assessment, 1998.

[7] Y. Wang & G. King, *Software Engineering Processes: Principles and Applications*, (CRC Press, 2000).

[8] P. Kuvaja, J. Simila, L. Krzanik, A. Bicego, G. Koch & S. Saukonen, *Software Process Assessment and Improvement: the BOOTSTRAP Approach*, (Blackwell Publishers, Oxford, UK, 1994).

[9] Y. Wang & A. Bryant, *Process-Based Software Engineering: Building the Infrastructures*, (Springer Netherlands Publishers, 2002).

[10] Correlation Ratio, MathWorld Team © 1999-2010 - Wolfram Research, Inc. http://mathworld.wolfram.com/CorrelationRatio.html.
[11] A.J. Walker and C. Gee, ISO 9001 Model Support for Software Process Assessment, *Journal Logistics Information Management*, Volume 13 , 2000, 39 – 44.
[12] W. Emmerich, *Software Process - Standards, Assessments and Improvement*, (Publisher Springer US Heidelberg USA, 1997).
[13] Automotive SIG, *Automotive SPICE Process Assessment Model*, The SPICE User Group 2005-2007. Online Posting http://www.itq.ch/pdf/AutomotiveSPICE_PAM_v23.pdf.
[14] F. Bomarius, H. Iida, Product Focused Software Process Improvement, (Springer, 2004).
[15] M.S. Krishnan and T. Mukhopadhyay *Software Process Models and Project Performance*. (Information Systems Frontiers, Kluwer Academic Publishers, Boston, 2000).
[16] H. Sahily. *A Comparative Study of Software Engineering Process Models*. Master Project. Lebanese American University. 2010.