

# Issues in the Implementation of Software Process Improvement Project in Malaysia

MOHD HAIRUL NIZAM MD. NASIR, RODINA AHMAD, NOOR HAFIZAH HASSAN

Department of Software Engineering  
Faculty of Computer Science and Information Technology  
University of Malaya, 50603 Kuala Lumpur,  
MALAYSIA  
hairulnizam@um.edu.my, rodina@um.edu.my

*Abstract:* Software Process Improvement (SPI) became known in the last twenty years. SPI is crucial to augment software process capabilities in software companies to face present demanding and global market. There were numerous published studies in United States, Europe, Australia and North America. Yet, there was still being short of research and published studies on SPI in Malaysia. This research attempts to fill this gap by focusing to analyze the resistance factors that de-motivate the implementation of SPI project specifically software companies operated in Malaysia. This research has been conducted during March 2008 until August 2008 and it used a survey instrument to gather data from 39 companies operated across Malaysia with the total of 251 professionals responded. The findings showed that organizational factors specifically human factors are playing an important role in determining the success of the SPI project. Participation and commitments from all individuals across the organization are also vital and imperative to ensure success for SPI initiative.

*Key-Words:* Software Process Improvement, Resistance Factors, SPI Implementation, SPI in Malaysia

## 1 Introduction

Software engineering community has increasingly aware that high-quality software development processes will produce a high quality product. This understanding is being inspired by the efforts of Deming [1] and Juran [2]. In another words, 'the quality of a product is largely governed by the quality of the process used to build it' [3]. A comparative study done by Hussain et. al [4] showed that software project tends to fail on many reason that can actually be handled with an effective management. With this understanding, many software researchers and practitioners have refocused their efforts on the process dimension of software engineering. Various software process improvement (SPI) models such as Software Process Improvement and Capability determination (SPICE) [5], BOOTSTRAP [6], ISO 9000[7] and the Capability Maturity Model (CMM) [7] have been proposed to assist organizations to achieve more predictable results by incorporating proven standards and procedures into their software process. Organizations that have made use of these standards advocated in ISO 9000 and CMM have usually shown excellent improvements. For example, by "improving its development process according to CMM 'maturity', Hughes Aircraft improved its productivity by 4 to 1 and saved

millions of dollars"[8]. This reported achievement has essentially motivated further the application and establishment of SPI initiatives in software organizations.

Brietzke and Rabello[3] conducted a survey in Brazil to identify resistance factors influencing the implementation of a software process improvement projects. Essentially 36 questionnaires from 18 companies were collected and analyzed. The research contributes in identifying main resistance factors perceived as critical to the implementation of SPI. Since the survey has been conducted only in Brazil, it is obviously useful to replicate the study in other part of the world. The replication will aid to verify the factors as well as to compare whether there is any major difference in other country such as Malaysia. Subsequently, this research abstracts the identified factors from Brietzke and Rabello[3]. It surveys 39 software companies which have been involved in SPI projects for more than two years. This works is also a continuation from previous survey of 29 companies which have been conducted in 2007. We increase the survey to 39 companies and get a better response from the extra 10 companies. 4 to 8 questionnaires were distributed to each company with the objective of getting wider views on each SPI project. The questionnaires were mainly distributed

to software professionals who are directly involved in SPI projects.

This paper intends to report on the findings. It is divided into 7 sections. The first section provides the introduction to the importance of having standards in software process improvement project. The second section gives a brief overview of the various software process improvement models used by software based companies in Malaysia, while the third section discusses a set of hypothesis in relation to resistance factors in the implementation of the SPI. The fourth section states the methodology used in conducting this research. The fifth section demonstrates the analysis of the survey result, while the sixth section provides ranking for resistance factors based on the analysis made on the previous section. The last section summarizes the main points of this research.

## 2 Software Process Improvements

The processes for large-scale software development can themselves be quite large and complex and may involve many software engineers, programmers and designers. As a result, these processes are frequently hard to define, difficult to understand, and sometimes even laborious to establish or standardize. Software process improvement (SPI) emerges to tackle these process-related difficulties. The underlying assumption behind SPI and the many SPI frameworks is that quality process will influence the quality of product. What is required for a quality process or a disciplined process is when it specifies the set of rules that would result in behaviour consistent with those rules. For a process to be effective, people must know about it and must be trained in it. In another words, to produce a quality or disciplined process, it has to be defined, trained and enforced until it become mature and is continuously improving from time to time (Humphrey 1989).

- *Software process* – “...a set of partially ordered process steps, with sets of related products, human and computerized resources, organizational structures and constraints, intended to produce and maintain the requested software products” [9].

At present, software models for improving the quality of software through management of the

software process have become significant in the software industry. Many companies are now being assessed according to de facto standards such as the Capability Maturity Model (CMM), SIX SIGMA or International Standards Organization (ISO) 9000.

Organizations that make use of the standards advocated in CMM, PSP, CMM-I, ISO usually show excellent improvements. For example, Irigoyen et. al [10] reports that BL Informatica was able to achieve a higher Cost Performance Index(CPI) of nearly 0.95 compared to its previous CPI of 0.38 before the SPI initiative. Ferguson *et al.* [11] claims that there is a schedule estimation improvement and strong quality improvements in the developed software when software engineering groups from three different companies, namely Advanced Information Services, Motorola and Union Switch and Signal use Personal Software Process (PSP) as their software process improvement model. In the case of IBM Australia Application Management Services, its on-budget delivery was improved from 90 percent to nearly 100 percent as it moved from SW-CMM maturity level 3 to CMMI maturity level 5. Siemens Information Systems Ltd also is able to reduce cost of poor quality from over 45 percent to fewer than 30 percent over a three year period as it moved from SW CMM maturity level 5 towards CMMI maturity level 5. These are some of the success stories behind SPI implementation initiatives.

Table 5 in Appendix A summarizes a spectrum of the most common SPI standards used by software based companies in Malaysia.

## 3 Resistance Factors in Software Process Improvements Projects

On the other hand, there are many difficulties and problems which have been reported by companies undergoing SPI projects. There are various factors which are influencing the implementation of SPI projects. Amescua et. al in the paper list out some of the resistance factors which apply for the organization who implements the SPI [12]. Meanwhile, Sakamoto et. al [13] has essentially identified main problems faced in most SPI implementation which include fluctuating goals, no visualized status of SPI project, poorly managed information, unclear role distribution and hardly transferred technology. On top of that, Arshad et. al [43][14] shows that effective communication in software project development environment among the

developers, users and project managers are the deciding factor to minimize project failure.

Using these identified problems and Beecham's research[15] as well as Brietzke and Rabello[3] work as the base, this research has extracted resistance factors which can be categorized into two main categories which are 1) Organizational factors and 2) Project factors. The organizational factors related to problems within the scope of the organization and are usually under senior manager's responsibility as reported in [15][16] and [17], meanwhile project factors related to problems regarding the software project management such as planning activities and resource distribution among others [18][19].

### 3.1 Organizational Factors

Software processes play an important role in coordinating different teams in large or small organizations so that their practices don't grow out and affecting with one another [20]. These are the factors which are related within the scope of the organization and are usually under senior managers' responsibility as presented by Brietzke and Rabello[3] and others in [21],[22],[23],[24],[25],[26]. There are five factors which are categorized under organizational factors namely 1) human 2) political 3) cultural 4) goals and 5) change management. Table 6 provides a description states the key resistance factor for each one as in Appendix B.

### 3.2 Project Factor

Project factors reflects the resistance facets on ongoing project which contributes and gives impact during the software project implementation stage. Hence, project factors include contributing elements in the projects as described by Wiegers [18] and others [2],[19],[27],[28],[29],[30]. There are 4 factors which are categorized under project factors namely 1) budget and estimates 2) documentation 3) quality and 4) tools and technologies. Table 7 in Appendix C provides a description of the key resistance factors.

## 4 Research Methodology

There are several stages on conducting this research which aim to replicate the survey performed by Brietzke and Rabello [3]. The focus

is on verifying the relevance of the identified resistance factors as well as comparing the similarities and differences between the studies. Essentially, we perform the initial literature review on SPI, looking at the broader context of SPI, key success factors and the difficulties highlighted in SPI projects. To obtain the overall picture of software process improvement standards, subjects related to the software process aspects, software qualities and software process improvement standards are also reviewed. Most of the material were collected using on-line search via the internet specifically on the online databases namely ACM, IEEE, technical reports published by Software Engineering Institute (SEI), academic textbooks, magazines, online articles and others.

Next, we focus on abstracting key resistance factors from Brietzke and Rabello and devising the questionnaires [3]. The resulting questionnaires were distributed to software companies which have been undergoing software process improvement project. 39 companies have been identified around the states of Kuala Lumpur and Selangor in Malaysia. From of questionnaires that have been distributed, 251 of professionals responded. The analysis of this survey is presented in Section 5. Section 6 illustrates the ranking of the surveyed resistance factors.

## 5 Research Findings

Survey results are described according to the section divided below:

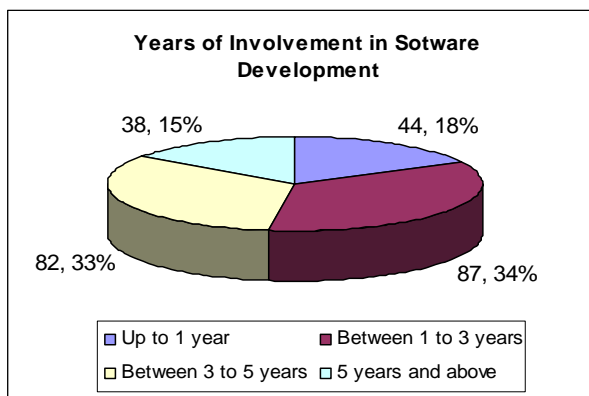
### 5.1 Demography Information

This section presents an analysis of the respondents' profile and companies taking part in this survey. The demography information has been placed in the first section in the questionnaires in the form of several multiple choice questions. The respondents' profiles captured in this first section include role in the organization, education level, and academic area, working duration in software development area, period of time working in SPI project and expertise level in the area of SPI.

Table 1 shows number of respondents according to their roles in the organization.

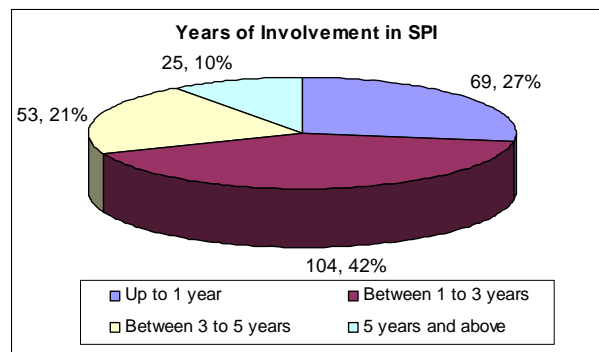
**Table 1: Number of respondents according their roles in their organization**

Roles	Numbers
Business Person	19
Project Manager / Quality Manager	22
It Consultant	20
System Analyst	50
Software Engineer /Developer	93
System Administrator	7
Designer	12
Others	28

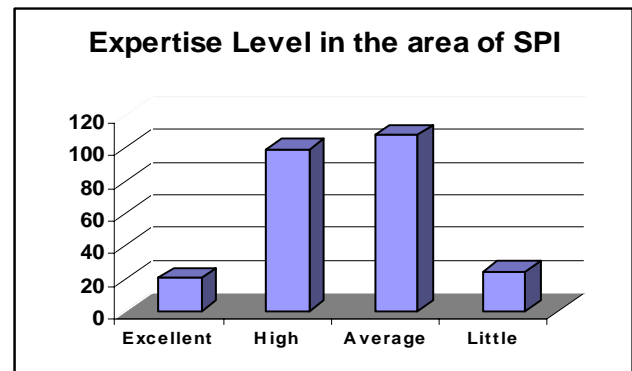


**Fig. 1: Number of respondents according to years of involvement in software development area**

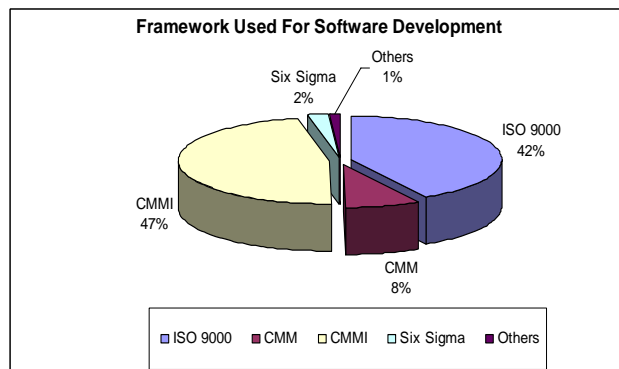
Fig. 1 shows the years of involvement of respondents in software development area, meanwhile, Fig. 2 demonstrates the years of involvement of respondents in the SPI project. Respondents were also asked regarding their level of expertise in the SPI using ordinal scales which start from *little* up to *excellent* level as illustrated in Fig. 3. The data shows that 108 respondents perceived themselves to have average level of expertise, 99 rated themselves to have high level of expertise and 20 respondents rated themselves to have excellent level of expertise. Overall, the data indicates the respondents have a good and sufficient knowledge of SPI and can provide reliable input to this survey.



**Fig. 2: Period of time working in software process improvement project**



**Fig. 3: Expertise level in software process improvement**



**Fig. 4: Framework used for Software Development**

Fig. 4 shows the frameworks or models used by the 39 companies in achieving the software process standards. The figure shows that the most used framework is the CMMI followed by ISO9000. It is quite difficult to explain the popularity of CMMI in Malaysian context. One of the possible reasons might be due to the flexibility offered by CMMI in terms of the selection of key process areas to be implemented. The high level deployment of ISO 9000 partly is because of the support of national government. Many of the government initiatives in Malaysia use ISO as their quality management model for improvements.

The Malaysian government also drives many public universities to achieve ISO certification. This trend has one way or another provides the awareness towards the greater use of the model. Other than that, there are no specific reasons for the high deployment of CMMI and ISO 9000 compared to other SPI models.

### 5.2 Resistance Factors

Sections II in the questionnaires cover the resistance factors that may influence and contribute to the delay or failure for the implementation of SPI. The questionnaire uses the scale of 1 to 5, ranging from the least influential to the highest influential factor. Ordinal scale is used whereby the user need to choose to rate the influence level of resistance factor from 1 until 5.

#### 5.2.1 Total Influence Level for the Organizational Resistance Factors

Table 3 summarizes the total of influence level score for each organization resistance factor according to the formulae below:

$$T(f_n) = \sum_{i=1}^n R(f_n).W(f_n) \quad (1)$$

$T(f_n)$  is the total of influence level score attributed to factor (f). It is a sum of the score rated by the respondent multiplied by weightage score according to influence level.

$R(f_n)$  is the score attributed to factor (f) as rated by the respondent according to the influence level

$W(f_n)$  is the weightage score attributed to factor (f) according to influence level as assigned in Table 2  
 $f_n$  is refers to the factor number.

**Table 2 : Value of influence level and weightage score**

Influence Level, <i>R</i>	Weightage Score, <i>W</i>
1	1
2	2
3	3
4	4
5	5

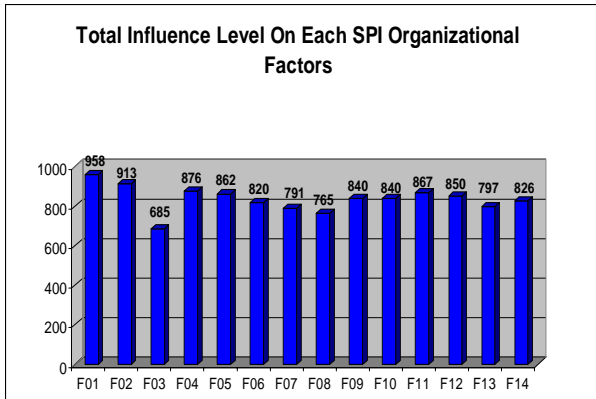
**Table 3: Total influence level in SPI area for the organizational resistance factors**

	Influence Level					Total of Influence Level Score, <i>T</i>
	1	2	3	4	5	
F01	9	21	43	47	54	958
F02	5	17	42	68	42	913
F03	8	24	57	57	28	685
F04	4	31	40	66	33	876
F05	2	12	75	64	21	862
F06	6	35	59	56	18	820
F07	9	45	56	43	21	791
F08	11	38	63	53	9	765
F09	4	32	61	56	21	840
F10	10	24	61	56	23	840
F11	3	26	67	55	29	867
F12	3	26	55	75	15	850
F13	4	31	80	45	18	797
F14	6	22	62	66	18	826

#### Legend:

- F01 Lack of commitment in all levels of the organizations
- F02 Lack of adhesion and participation of all the individual involved in SPI projects
- F03 Lack of professionals experience and skill
- F04 Lack of leadership and backup by top management level
- F05 Lack of adequate training
- F06 Lack of the establishment of organizational policies.
- F07 Lack of the establishment of Quality Policy
- F08 Lack of expertise in implementing cultural changes.
- F09 Lack of consistency between software processes improvement project and the organization's strategic objectives
- F10 Absence of focus on the organization's most urgent needs.
- F11 Unrealistic expectation towards the SPI project.
- F12 Insufficient and ineffective assessment of the current software process
- F13 Existence of a software processes improvement project team not focused on orientation and technical support.

F14 Simultaneous focus on many improvement areas



**Fig. 5: Total influence level score according to organizational resistance factors**

All the results gained in Table 3 are then plotted in a graph format as illustrated in Fig. 5. It can be observed that the top 3 organizational resistance factors are factor number 1 which is *lack of commitment in all levels of the organizations*, followed by factor number 2 which is *lack of adherence and participation of all the individual involved in SPI project*, and the third one is factor number 4 which is *lack of leadership and backup by top management level*. These top 3 resistance factors are categorized under people factor as referred in Table 3. The bottom three (lowest ranking) of organizational resistance factors are *lack of professionals experience and skill*, *lack of expertise in implementing cultural changes* and *lack of the establishment of Quality Policy*.

### 6.2.2 Total Influence Level for the Project Resistance Factors

Table 4 summarizes the total of influence level score for each project resistance factor according to the formula given in section 5.2.1. Then, the results gained are plotted as illustrated in Fig. 5 below

**Table 4: Total influence level score according to project resistance factors**

	Influence Level					Total of Influence Level Score, <i>T</i>
	1	2	3	4	5	
F15	7	20	49	57	41	832

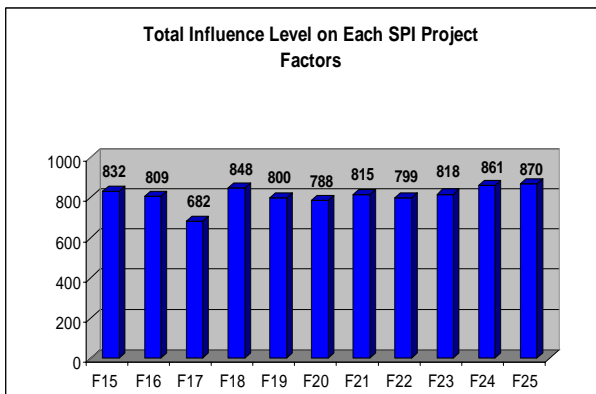
F16	6	30	61	54	23	809
F17	3	24	55	70	22	682
F18	6	20	61	54	33	848
F19	9	36	52	54	23	800
F20	5	28	82	52	7	788
F21	8	23	65	56	22	815
F22	7	36	55	63	13	799
F23	8	27	69	51	19	818
F24	4	23	62	64	21	861
F25	0	22	70	59	23	870

**Legend:**

- F15 Current budget and estimates exceeds planning
- F16 Lack of understanding by top management level that the software processes improvement project is a long-term return on investment process.
- F17 Lack of visibility about the ongoing software processes improvement project activities.
- F18 Excessive documentation and formality.
- F19 Lack of infrastructure and of a documentation management.
- F20 Lack flexibility in the use of the documentation in projects of different types and sizes.
- F21 Lack of involvement of top management in the relationship between the project teams and the person or group of quality assurance.
- F22 Lack of treatment to guarantee process conformity in instances of hiring and/or dismissal of skilled professionals.
- F23 Automation of not well-defined processes
- F24 Lack of training on the support tools and technologies defined as support.
- F25 Pressure and absence of planning concerning the adaptation period.

As illustrated in Fig. 5, the total of influence level score for each of the project resistance factor is very close to each other with standard deviation of 50.29. This finding is not much different in comparison with organizational factors. The top 3 resistance factors in the category of project factors are: *pressure and absence of planning concerning the adaptation period* (factor 25) followed *lack of training on the support tools and technologies* (factor 24) and the third rank is *excessive documentation and formality*

(factor 18). Meanwhile, the three lowest of project resistance factors is *lack of visibility about the ongoing software processes improvement project activities (factor 17)*, followed by lack of flexibility in the use of the documentation in projects of different types and sizes (factor 20). The third lowest is *lack of treatment to guarantee process conformity in instances of hiring and/or dismissal of skilled professionals (factor 22)*.



**Fig. 6: Total Influence level Score according to Project Resistance Factors**

## 6 Consolidation of the Results

Based on the survey findings reported in Section 5, total influence level score for both organizational and project resistance factors are being merged to gain overall results in order to determine the most and the least influence resistance factors.

According to the results survey, the most critical resistance factor is *lack of commitment in all levels of the organizations*. This result is not similar with the result gained by Brietzke and Rabello [3] and corroborates the research findings experience in SPI projects. This factor is directly influenced by the size or hierarchy of the company. It appears that the larger the size or bigger hierarchy in a company, more time is needed to get full commitment from all levels of the organization. The second most critical resistance factor from the finding is lack of adherence and participation of all the individual involved in SPI projects. The third most critical resistance factor is lack of leadership and backup by top management level. It is essential that clear expectations and goals need to be specified very early so that progress towards those goals can be continually monitored so that

revisions to either goals, of processes, or both can be made persistently.

All these top 3 resistance factors are classified under organizational factor as described in details in Section 3.1. The results obtained in this research is not much different with the results survey conducted by Brietzke and Rabello[3] whereby both human factors which are lack of adherence and participation of all the individual involved in SPI projects and lack of commitment in all levels of the organizations give a big influence in determining the success of the SPI projects. Moreover, these three factors which have been perceived to be critical are obviously considered to be the most difficult elements which can be expected from organizational members. Participation, commitment and reasonable expectations are the end result which should be manifested by the organizational members, if they are willing to contribute to the SPI project. This survey finding strongly advocates that organizations implementing SPI projects should spend more effort to create awareness and gain full participation and commitment from their staff to ensure successful implementation of SPI project.

Meanwhile the 3 least resistance factors identified in this survey which are lack of visibility about the ongoing software processes improvement project activities, lack of professionals experience and skill and lack of expertise in implementing cultural changes. These 3 factors might have been considered less critical due to the background of most of the respondents. 42% of them have at least 1 to 3 years experience in SPI projects. Based on these, they might perceive that having experience, expertise and vision are not the most critical components which may contribute to the successful implementation of SPI. In contrast, one of the top most resistance factors identified in survey performed by Brietzke and Rabello[3] is lack of expertise in implementing cultural changes. The difference in the perceived importance of this factor might be due to the background of the respondents.

On top of the existing survey question, we have also received interesting comments from the respondents. Several respondents revealed that the progress of SPI project implementation may get badly affected if SPI schedule mix up with the ongoing software development project in their companies. The respondents are suggesting that proper and synchronized planning should be done to ensure that the SPI implementation schedule can be carried out harmoniously with the ongoing software development project.

## 7 Conclusion and Future Research Directions

This paper has identified and analyzed crucial resistance factors which influence the implementation of the software process improvement project specifically in companies operated in Malaysia. The top three and the bottom three least resistance factors have been listed in Section 6. It concludes that organizational factors specifically human factors are playing vital roles in determining the success of the SPI projects. Moreover, participation and commitments from all individuals across the organization are crucial to ensure successful implementation for SPI initiative.

One obvious limitation of this study is in using only one set of questionnaire and using only the survey instrument to gather information. In this case, the information gathered is limited to the questions asked. Deeper understanding may be gained if other research methods are used in combination. Our future work intends to increase the number of participating companies and uses additional data gathering techniques with the objective of getting wider and more accurate picture of the SPI implementations.

There are other interesting issues in SPI that can be ventured into as well. The issues include comparing the difficulties or obstacles faced by big, medium and small software companies in initiating or deploying SPI projects. Due to the popularity of CMMI deployment in Malaysian companies, we are also interested to actually understand further the implementation of CMMI and the reason companies are interested with CMMI.

Finally, our main aim in this endeavor is to facilitate other software companies to consciously manage future projects through the use of protective actions or proper planning which can reduce the anticipating problems during SPI projects implementation.

## References

- [1]Deming, W. E. *Out of the Crisis*, MA: MIT Press, Cambridge, 1986.
- [2]Juran, J. M. and Gryna, F. M., *Juran's Quality Control Handbook*, Fourth ed., New York: McGraw-Hill Book Company, 1988
- [3]Brietzke, J. and Rabello A., Resistance Factors in Software Process Improvement, *CLEI Electronic Journal*, Vol.9, 2006
- [4]Hussain, S. J., Rhashid, K., Ahmad, H. F., Hussain S.F., Effective Software Management– Where Do We Falter? , *Proceedings of the 6th WSEAS Int. Conf. on Software Engineering, Parallel and Distributed Systems*,2007, pp. 13-17
- [5]ISO/SPICE, 2008. URL <http://www.isospice.com/categories/SPICE-Project/15/03/2008>
- [6]Russell, C. and H. Cheng, Bootstrap Methods in Computer Simulation Experiments, *Winter Simulation Conference*, 1995, pp. 171 - 177.
- [7]Carnegie Mellon, Software Engineering Institute, “*Capability Maturity Model Integration (CMMI)*”, Version 1.1, 2002.
- [8]Pfleeger, S. L. *Software Engineering: Theory and Practice*, Upper Saddle River, NJ: Prentice Hall, 1998.
- [9]Lonchamp, J. ,A Structured Conceptual and Terminological Framework for Software Process Engineering, *Second International Conference on the Software Process*, IEEE-CS Press, Los Alamitos, CA, USA.,Oct. 1993, pp 41-53.
- [10]Irigoyen Ferreiro, A., Santos G. , Cerqueira R. , Montoni M. , Barreto A., Andrea O.,Barreto S., Regina Rocha A. ,”Applying ISO 9001: 2000, MPS.BR and CMMI to Achieve Software Process Maturity: BL Informatica's Pathway, *International Conference on Software Engineering*, 2007, pp. 642-651
- [11]Ferguson, P., Humphrey, W. S., Khajenoori, S., Macke and Matvya, S. A., Introducing the Personal Software Process: Three industry cases, *IEEE Computer*, Vol. 30, No.5, 1997, pp. 24–31.
- [12]Amescua, A.,García, J., Sánchez-Segura, M. and Medina-Domínguez, F., Software Process Improvement for Practitioners Based on Knowledge Management Tools, *Proceedings of the 5th WSEAS Int. Conf. on Software Engineering, Parallel and Distributed Systems*,2006, pp 24-29.
- [13]Keishi Sakamoto, Kumiyo Nakakoji, Yasunari Takagi, Naoki Niihara,Toward computational support for software process improvement activities,*International Conference on Software Engineering*, 1998, pp 22-31.
- [14]Arshad, N.H., Mohamed, A. and Mat Nor, Z., Risk Factors in Software Development Projects *Proceedings of the 6th WSEAS Int. Conf. on*



*Software Engineering, Parallel and Distributed Systems*,2007. Pp.51-56.

- [15]Beecham, S., Tracy H. and Austen R.,Software Process Improvement Problems in Twelve Software Companies: An Empirical Analysis, *Empirical Software Engineering*, Vol.8, No.1, 2003, pp. 7-42.
- [16]Kitchenham, B. A, Evaluating Software Engineering Methods and Tool's, *ACM SIGSOFT Software Engineering Notes*, 1996.
- [17]Abrahamsson, P. Commitment Development in Software Process Improvement: Critical Misconceptions, *Software Engineering*, 2001, pp.71-80.
- [18]Weigers, Karl E. Software Process Improvement: Ten Traps to Avoid ,*Software Development*,1996
- [19] Statz, J., Oxley, D., O'Toole, P. Identifying and Managing Risks for Software Process Improvement, 2007.
- [20]Tore Dyba, Factors of software process improvement success in small and large organizations: an empirical study in the scandinavian context,*ACM SIGSOFT Software Engineering Notes*,Vol. 28, No. 5, 2003, pp. 148-157.
- [21]Kumlander,D., Personnel motivating projects: reasons, implementation and risks *Proceedings of the 6th WSEAS Int. Conf. on Software Engineering, Parallel and Distributed Systems*,2007,pp. 30-34.\
- [22]Wheeler, S. and Duggins, S., *Improving Software Quality*, Southeast regional conference, 1998,pp.300-339.
- [23]Taylor, D. and McGraw, G. Adopting a Software Security Improvement Program, Vol.3, No.3, 2005.
- [24]Lazic, L. and Mastorakis, N., Cost Effective Software Test Metrics, *WSEAS Transactions on Computers*, 2008, Vol 6, No.7, pp. 559-619
- [25] Clarke, L A. and Osterweil, L. J.,Continuous Self-Evaluation for the Self-Improvement of Software, International workshop on Self-adaptive software, 2000, pp. 27- 39.
- [26]Miler, J. and Górski, J., Risk-driven Software Process Improvement - a Case Study, *EuroSPI'2004*, 2004.
- [27]Ojala, P., Towards a Value-Based Approach in Software Engineering, *2nd WSEAS Int. Conf on Computer Engineering and Applications (CEA'08)*, 2008, pp.54-59.
- [28]Ojala, P. Experiences of Implementing a Value-Based Approach. *WSEAS Transactions on Information Science & Spplications*.Vol.5,No 4. 2008. pp. 385-395
- [29]Lazić, L. And Dakića, R. The Software Testing Challenges and Methods, 9th *WSEAS International Conference on Communications*. 2007, pp. 497-804.
- [30]Umarji, M. and Seaman, C., *Predicting Acceptance of Software Process Improvement*, 2005.
- [31]Paulk, M. C. The Evolution of SEI's Capability Maturity Model for Software, Software Engineering Institute, CMM Evolution, 1994, , pp. 1-71
- [32] Herbsleb, J., Carleton, A., Rozum, J., Siegel, J. and Zubrow, D. CMM, Benefits of CMM-Based Software Process Improvement: Initial Result, 1994.
- [33]Paulk, M., Charles, C., Weber, V., Curtis, B. and Chrissis, M. B. *The Capability Maturity Model: Guidelines for Improving the Software Process*, Addison Wesley, Carnegie Mellon, Software Engineering Institute, 1995.
- [34] Kumar, G. P. Software Process Improvement – TRIZ and Six Sigma, (Using Contradiction Matrix and 40 Principles), 2005.
- [35]Mccarthy, B. and Stauffer R., Enhancing Six Sigma Through Simulation With Igrafx Process For Six Sigma,*Winter Simulation Conference*, 2001, pp 1241-1247.
- [36]Knuth, K., Niebuhr, G., Lind, M. amd Mauer, B. Six Sigma – Making Corporations and Stockholders,*Industrial Engineering 361*, 2002.
- [37]Kermani Amir H. M. Empowering Six Sigma methodology via the Theory of Inventive Problem Solving (TRIZ), 2003.
- [38]Hurber, C. and Mazur, G. H., *DFD and Design for Six Sigma*, symposium on DFD, 2000.
- [39] Ashok Shenvi A., Design for six sigma: software product quality, *Proceedings of the 1st conference on India software engineering conference*,2008 , pp. 97-106.
- [40]Galín D., *Software Quality Assurance: from theory to implementation*, Addison Wesley, United States, 2003.
- [41]Paulk, M. C., A Comparison of ISO 9001 and the Capability Maturity Model for Software, Software Capability Maturity Model Project, 1994, pp. 1-71.

## APPENDIX A

**Table 5: Summary of Various Software Process Improvement Standards**

	<b>Description</b>	<b>Implementation Method</b>
<b>CMM</b>	According to Paulk <i>et al.</i> in [31] reported that CMM guides software groups on how to gain control of their processes for developing and maintaining software and how to evolve toward a culture of software engineering and excellence of management. It provides a framework for organizing these evolutionary steps into five maturity levels to act as a successive foundation for continuous process improvement. The five levels as defined by are initial, repeatable, defined, managed and optimized as mention by Herbsleb <i>et. al</i> in [32]	As described by Herbsleb <i>et. al</i> [32] each level in CMM needs to reach the Key Process Area (KPA) which consists of key practices that contribute to satisfying its goals. Each Key Area must implement the pre-established goals, which are, activities to be developed, or necessary infrastructure to the goals satisfaction. The key practices are aggregated can either be the implementation or Institutionalization. Each maturity level establishes a different component in the software process, resulting in the capability increase of the organization process.
<b>CMM-I</b>	Software Engineering Institute (SEI) describes that the Capability Maturity Model Integration (CMM-I) as an SPI process which provide a guidance for improving on each organization's processes and the capability to control and manage the development, achievement and maintenance of products or services during software process [33] It is a staged representation, organizes process areas in five maturity levels which is same as CMM. The different is that, CMM-I is an integrated approach which establish a framework to integrate current and future models or build an initial set of integrated models.	In order to achieve the CMMI, the organization must select the models by choosing either continuous or staged representation [33]. Continuous refers to an organization to select the order of improvement that best meets the organization's business objectives and mitigates the organization's areas of risk. Meanwhile representation staged helps by providing a proven sequence of improvements, starting with basic management practices and progressing through a predefined and proven path of successive levels, each serving as a foundation for the next.
<b>SIX SIGMA</b>	Six Sigma is quality program pioneered by Motorola invented by Bill Smith who is senior engineer at Motorola whereby the main objective is to reach a quality goal on their products [34]. It emphasizes on a good financial results that can be achieved through the virtual elimination of product and process defects [35]. Six Sigma follows a sequence of process which involving 5-steps, known as DMAIC (Define, Measure, Analyze, Improve and Control) as mention in which each steps improves to satisfy the software process [35].	Two implementation methods involve in Six Sigma as mentioned by Knuth <i>et. al</i> are the most well-defined if a problem with an unknown solution existing products, processes or services which method is DMAIC [36].( Define, Measure, Analyze, Improve and Control) The latest method, which still in the developing stage is called Design for Six Sigma (DFSS) as described by Kermani [37]. The goal of DFSS is to develop a new product, process or service that is eliminate the defects. However, a few of consulting companies have invented roadmap for DFSS like IDOV (Identify, Design, Optimize and Validate) as mention by Hurber and Mazur in [38] and DMADV (Design, Measure, Analyze, Design and Verify) by Knuth <i>et. al</i> [36]. A paper described by Ashok in [39] introduces "DFSS" (Design for Six Sigma) methodology to boost product quality which can improve software process.

<b>ISO-9000</b>	<p>It is a set of standards, adopted by the International Organization for Standardization (ISO) based in Geneva, Switzerland which defines the requirements for an organization's Quality Management System (QMS) as mentioned by Galin [40]. Paulk describes ISO 9000 as a quality management and assurance standards which provides guideline for selection and use; clarifies the dissimilarities and interrelationships between quality concepts and provides guidelines for the selection and use of a series of international standards [41]. ISO-9000 can be divided into three main components which is the first one is a framework that describes the overall requirements for the QMS. Secondly, a life-cycle activity which covers requirements for all phases of the development process until software maintenance and finally supports activities which identifies requirements for sustaining all the actions.</p>	<p>According to Galin [40], for any organization that is requesting certification, initially the organization need to prepare planning processes which require them to have a proper development of organization that fulfills the SQA system. The certifying organization will undergo to the certification audits which include review the quality manual and SQA procedure developed by the organization .If those criteria comply with ISO 9000 ,the performance audit of management system take place. If its not, the organization need to refine back the organization's SQA system. Meanwhile, the implementation of organization's SQA system takes place. Followed by the performance of SQA management system is checked whether its comply or not with ISO 9000.If all the criteria is fulfill ,the ISO-9000 is issue and if its not the organization need to carry out the performance improvements of organization of SQA management system again.</p>
-----------------	---	--

## APPENDIX B

**Table 6: Description of Each Organizational Factor**

	<b>Description</b>	<b>Key Resistance Factors</b>
<b>Human</b>	<p>According to Abrahamsson without commitment from all organizational levels (human) to support SPI, the initiative will most likely fail or the results are not far reaching [17]. The experience of senior management with an SPI project will give positive impacts to the improvement process. Consultation support such as advice and training of SPI action teams and staffs is one critical aspect in ensuring the success of SPI project. Beecham <i>et al.</i> in stated that organizational issues (especially the human element) are important contributing factors to the success of SPI initiatives [15]. Supported by Kumlander [21],that emphasizes on how the organizational plays an important role in the personnel motivating projects which is one of the factors that effecting the successfulness of the company</p>	Lack of commitment in all levels of the organizations
		Lack of adhesion and participation of all the individual involved in SPI projects
		Lack of professionals experience and skill
		Lack of leadership and backup by top management level
		Lack of adequate training
<b>Political</b>	<p>According to Wheeler and Duggins the political factor is important to the construction for a department to SQA [22]. The establishment of quality policy which is one of the political issues comes after the commitment of senior management. Quality policies and standards for SPI efforts describe the organizational goals and objectives related to the quality.</p>	Lack of the establishment of organizational policies
		Lack of the establishment of Quality Policy

<b>Cultural</b>	Taylor and McGraw proved that in order to ensure success in a cultural change program, a champion who can build, deploy, drive, and own each initiative going forward must be properly decided [23]. However, every cultural change program requires good cooperation from both management and tactical technical staff; improvement programs will fail if either group is left out or underemphasized.	Lack of expertise in implementing cultural changes.
<b>Goals</b>	According to Wiegers's research, if the goals, deadlines, and expected results by the managers are impractical, the effort towards SPI may be unsuccessful [18]. As mentioned by Lazic and Mastorakis [24] 'software development is measured in terms of overall progress in meeting functional and business goals' that ensure how the software process is doing well. Meanwhile, Clarke and Osterweil found a reason why software process needs to have indefinitely ongoing processes [25]. It is essential that clear goals need to be specified earlier, so that progress towards those goals can be continually monitored, and so that revisions to either goals, of processes, or both can be made persistently.	Lack of consistency between software processes improvement project and the organization's strategic objectives Absence of focus on the organization's most urgent needs. Unrealistic expectation towards the SPI project.
<b>Change Management</b>	Initial analysis needs to be conducted to determine whether the SPI initiative apt with the organization's objectives and interests. This is also discussed and supported by Statz <i>et al.</i> in [19]. Wiegers in conceived that the SPI project's team be used to actively facilitate the efforts toward changes on the part of the project teams rather than simply check the situation of the ongoing process in order to report a long and depressing list of findings [18]. Miler and Górski highlighted that in order to have a successful software process improvements, risk from configuration & change management which is not explicitly defined will lead to unsuccessful business process [26].	Insufficient and ineffective assessment of the current software process Existence of a software processes improvement project team not focused on orientation and technical support. Simultaneous focus on many improvement areas

## APPENDIX C

**Table 7: Description of Each Project Factor**

	<b>Description</b>	<b>Key Resistance Factors</b>
<b>Budget and Estimates</b>	According to Wiegers lack of progress in improvement plans is frustrating to those who really want to achieve progress and this put down the importance of time and costs in the process evaluation [18]. Study conducted by Ojala[27][28]described that cost is the most difficult factor to handle in software development. Further research support by Statz stated that the senior management often overlook on the manageability of SPI project size issue and sufficient budget size for SPI project issue [19]. The first year is the most difficult	Current budget and estimates exceeds planning. Lack of understanding by top management level that the software processes improvement project is a long-term return on investment process Lack of visibility about the ongoing software processes improvement project activities.

	<p>period for a SPI program. Resistance to change will be at its peak, the costs are likely to be higher than in the 17 following years, and due to the steep learning curve, the first year goals and targets can easily be missed.</p>	
<b>Documentation</b>	<p>In SPI project, documentation is a must in order to provide proof and dissemination throughout the organization in a formal way. Therefore, it is helpful to have an infrastructure for documentation, since it is a mandatory practice throughout the organization. According to Beecham <i>et. al</i>, the documentation is also gaining importance in the list of problems associated to SPI[15]. It includes data measurement, proceedings register, coordination and management of the documentation, data collecting the operational framework forms the relationships and dependencies between what is to be done, by whom, and how to do it.</p>	Excessive documentation and formality.
		Lack of infrastructure and of a documentation management.
		Lack flexibility in the use of the documentation in projects of different types and sizes.
<b>Quality</b>	<p>Quality refers to the state of the software as it was released or delivered to customers. In SPI, quality involves a good team management explained by James <i>et. al</i> in his technical report ,is with a good procedural advice from professionals will lead to a good improvements of software quality [28]. In order to achieve higher level of quality assurance is by creating consent about how all the requirements definition processes has to be performed and which information should be provided to ensure the successfulness of the SPI project. ‘The increasing cost and complexity of software development is leading software organizations in the industry to search for new ways through process methodology and tools for improving the quality of the software they develop and deliver’[29].</p>	Lack of involvement of top management in the relationship between the project teams and the person or group of quality assurance.
		Lack of treatment to guarantee process conformity in instances of hiring and/or dismissal of skilled professionals.
<b>Tools &amp; Technology</b>	<p>The problem associated refers to the implementation of new tools and technologies, amount of work and pressures that hinder the use of new tools. According to Umarji , complexity of SPI tools and technologies need to ease with which developers can adapt to changes in work practices caused by SPI [30]. However it has several acceptance issues because it often involves learning new technology, changes in work practices and an additional workload. Also, SPI involves collecting data about projects, resources and deliverables and often practitioners are not keen on sharing this type of data</p>	Automation of not well-defined processes
		Lack of training on the support tools and technologies defined as support
		Pressure and absence of planning concerning the adaptation period.