Software Process Enhancement Model for SMEs

Babak Bashari Rad[†], Ammar AL-Ashmori[†], and Zahra Ahanin[††]
[†] Asia Pacific University of Technology & Innovation (APU), Technology Park Malaysia, Kuala Lumpur, Malaysia
[††] University of Malaya (UM), Kuala Lumpur, Malaysia

Summary
In the software industry, SMEs are important as they have vital contribution across the industry so it is compulsory for SMEs to assurance their software quality. In this context, many SMEs interested to enhance their software development process to achieve that. Regrettably, 82% of software companies face issues in adopting agile methods and practices, and most of SMEs fails to adopt CMMI successfully. Furthermore, SMEs have lack of experience on how plan the enhancement effort that makes the implementation of SPI fails. This paper propose a model to help SMEs to enhance their software process based on the suitable CMMI KPAs practices and agile practices and address the required efforts. Typical problems in enhancing SMEs software processes are addressed in some process patterns, which have been solved by the suitable agile practices, mapped to the equivalent CMMI KPAs practices.

Key words:
Agile, SPI, CMMI, SMEs, Software Process Enhancement Model

1. Introduction

Today, Small and Medium Sized Enterprises (SMEs) are widely recognised in making a vital contribution in software development process across industry. Considering the lack of universality of SMEs definition, its definition, and standards varies from one country to another. According to the definition of SME Corporation Malaysia, the sectors with employees from 5 to less than 30, fall into small enterprises, the medium size enterprise has 30 to 75 employees. However, European Union (EU) defines a small enterprise with size capacity from 10 to 49, and a medium size enterprise has 50 to 249 [1]. Giving the fact that, SMEs contribute to major economic activity throughout many nations worldwide [2, 3], it is necessary to make SMEs more efficient and effective organizations. Usually, for SMEs to improve their software and rise up the development productivity, SPI is recommended as the best way for affectivity [4]. By conducting SPI, these SMEs can continue and rise economic benefits as they improve the quality of their software process, which can cut down the cost, and time of building quality software products [5].
This study is aiming on proposing a process model to guide SMEs in adopting software development processes. The model addresses typical issues that SMEs facing to improve their software process to achieve a quality product. In this model, CMMI KPAs are mapped to the suitable practices from XP and Scrum. The model also includes critical success factors CSFs that are important to validate the implementations of this model.
In this paper, a short literature review about CMMI and agile and a discussion about the previous related works are presented, followed up by the research methodology. The proposed model is presented in the next section, followed up by the validation and this paper is closed by the conclusion and recommendations for future works.

2. Literature Review

The software Process Improvement (SPI) aims to increase the quality of the software products or service, and SMEs must follow SPI to maximize the economic benefits. Reason is that SPI can improve quality of their software process, increase readability, customer satisfaction and cut down the cost and time of building quality software products while reduces risk and fail [6]. Sommerville [1] believed that the implementation of SPI takes a chain of continuous and iterative enhancements to the current software process practices. These processes continuously change and improve as new practices can be added as well. Usually, SPI concentrates on the organization's software needs and the weakness of the current practices. In order to manage software development activities in organizations and achieve quality products, a number of software quality models have been introduced. Notable quality models are the Software Process Improvement and Capability determination (SPICE), Capability Maturity Model Integration (CMMI), and Six Sigma and ISO 9001 [7, 8].
CMMI has rapidly become a preferred instrument for process improvement around the word. It provides a single, integrated, and more agile framework for guiding and appraising improvement activities across an entire organization [9]. The CMMI can help the management to make right or appropriate forecasts of the business by supporting effective management and predictable schedules, which give value to the stockholders. In a study by [10], the researchers studied more than 400 projects in which the results show using CMM or CMMI based improved programs leads to 12% reduction of the overall lead-time and 49% reduction in defects.
Despite several advantages, many SMEs refused to adopt CMMI. Several researches establish that CMMI tends to match the requirements of large organizations and requires

Manuscript received July 5, 2017
Manuscript revised July 20, 2017

tailoring to suit SMEs needs [11]. Tailoring CMMI for the benefits of SMEs is not a trivial task, due to its complexity, the need for sufficient resources such as skilled professionals, challenging deadlines and high implementation costs, which is not affordable by SMEs firms [12, 13, 14].

In recent trend, defining a suitable approach for SMEs seems to be visible by taking into consideration some existing CMMI Key Process Areas (KPAs) and agile methods and practices. An increasing number of software organizations adopt CMMI using agile software development methods. According to CMMI Institute, in 2015 more than 70% of CMMI appraised organizations reported using one or more agile approaches [15, 16].

3. Previous Related Works

Developing the SPI framework as an alternative way for the CMMI can have the advantages of both the CMMI and agile methodologies. It makes them collaborate with each other in a unified way. Many researchers attempted to address and tackle these challenges by introducing with some SPI frameworks suitable for SMEs rather than CMMI.

There exist several empirical researches on developing a methodology by integrating CMMI with other SPI models. One such methodology is of integrating CMMI with XP, which covered all the software engineering practices in the 5 CMMI levels, and successfully mapped 30% KPAs with XP practices [17]. Lukasiewicz and Miler [18] merged CMMI with Scrum which covers most of the project management aspects in CMMI level 2 and 3 which is 60% of these 2 levels KPAs. The proposed model was evaluated with good results. Similarly, [19] combined CMMI and Scrum focusing on project management practices. [13] successfully blended CMMI with Six Sigma, and provided tools and templates to help SMEs reach the required CMMI level. M. A. Muñoz-Mata et al. [20] provided a set process pattern and web tool for facilitating the proposed framework, which took advantage of several agile methods. The framework covered software engineering as well as management aspects, and it was evaluated with acceptable results.

Though these studies have several advantages in reducing costs and improving quality, but still there is gap and limitations such as no or partially evaluation, lack of clarity in addressing CMMI KPAs and focusing only on one of the practices (software engineering practices, or project management practices, etc.). More details about the previous related works are explained on [1]

4. Research Methodology

This research follows the approach of research onion since it explains the layers of the research design to formulate an appropriate methodology [21]. Interpretivism paradigm, which goes along with inductive reasoning, is chosen because it helps to understand the social and cultural contexts that affect the SPI process like the software organizations and the people involved in software development process [22]. Additionally, this research design uses different kind of quantitative and qualitative methods to achieve different objectives. Literature review method helps to achieve the first objective, which is to study the SPI and the related aspects to it, in addition to, SMEs and their characteristics. The second objective, which is to investigate the suitable CMMI practices and agile practices for SMEs to improve the software development process, was achieved used literature review and a survey. The knowledge gathered from the first and second objectives helped to achieve the third objective, which is to propose this model, and called software process enhancement model (SPEM). Finally, the fourth objective, which is to find out the CSFs for validating and implementing the proposed model, was achieved using qualitative methods the focus group and Delphi method. The Delphi method is more suitable in situation where there is lake of data, and then it gathers opinions and thoughts of the experts [23].

4.1 Data collection approach (The survey)

Developing To address the second objective, the questionnaire survey method was selected. This objective is related to CMMI practices and agile in practices in SMEs. The population of this research includes managers, business analysts, quality assurance engineers, and developers in the company affected by SPI. The survey questionnaire developed online at Google Docs. It is free service, and convenient to create, share, and analyse the data. The link for the survey questionnaire was sent to 64 potential respondents in a company through email and using social media sources as Facebook and WhatsApp. The respondents were requested to fill in the online questionnaire. There were not specific criteria in selecting the respondents; they were selected randomly. Simple random sampling was used so all the individuals have a possibility to be chosen equally then the generalization can be done to the sample to larger population [24]. The data was gathered over a period of three months.

Out of total 64 requests sent to respondents, only 40 anonymous responses were received; therefore, the response rate was 62.5 percent. The respondents of this research were 4 project managers, 10 team managers, 1 manager associate, 2 business analysts, 3 senior developers, 17 developers, 2 quality assurance engineers,

and one person who did not specify his job. The collected data was analysed using statistical analysis method, which represent the survey data in bars, graphs, and pie charts. More details about the survey are on [25].

5. The proposed model

The purpose SPEM is to guide the company to identif y weaknesses in their current software process based on CMMI KPAs and how to implement the changes to the software processes. The core idea of this model is about a set of process patterns, which considered about the typical problems facing SMEs in improving their software process, which can be represented by CMMI KPAs and about mapping agile practice form XP and Scrum with these patterns as shown in Table 1.

The model contains three stages: analysing, planning and implementing. In analysing stage, there are four contexts, which represent CMMI maturity levels, and each context contains several patterns which represent the KPAs belong to that CMMI maturity level as show in Table 2. These patterns are considered as reusable units that the company can adopt to achieve the required maturity level. Based on the desired CMMI maturity level the context is set and its patterns are selected to address the improvement effort.

Table 1: Agile practices mapped to CMMI KPAs

	Patterns (CMMI KPAs)	XP practices	Scrum practices
Repeatable	2.1 Requirements management	user-stories, an On-Site customer	Product Backlog, Release Backlog, Scrum Board
	2.2 Project planning	planning game, small releases	Sprint Planning Meeting
	2.3 Project monitoring and control	big visual chart	Product Backlog, Release Backlog, Scrum Board, Daily Scrums
	2.4 Supplier agreement management		
	2.5 Measurement and analysis		Sprint Planning Meeting, Scrum Board, burn-down chart
	2.6 Process and product quality assurance	Pair programming	Sprint Review Meeting
	2.7 Configuration management	Planning game, collective ownership, small releases, continuous integration	
Defined	3.1 Requirements development	On-Site customer, user stories, iterative development	Sprint Planning Meeting, Sprint Review Meeting
	3.2 Technical solution	metaphor, iterative solutions, testing units	
	3.3 Product integration	planning game, iterative cycles, unit testing	Sprints, Sprint Review Meeting
	3.4 Verification	On-Site customer, user stories, iterative development	Sprint Review Meeting
	3.5 Validation	peer reviews, iterative development, unit test, On-Site customer	Sprint Review Meeting
	3.6 Organizational process focus	Team Focus	Scrum process itself
	3.7 Organizational process definition	Coding standards	Scrum process itself
	3.8 Organizational training	collective ownership	
	3.9 Integrated project management	planning game, visual charts, iterative developments	Sprint Review Meeting, Scrum of Scrums, Product Backlog
	3.10 Risk management	On-Site customer, XP-testers	Daily Scrum, Scrum of Scrums, Iterative approach limits risk
Managed	4.1 Organizational process performance		
	4.2 Quantitative project management		Estimation during Sprint Review Meeting, burndown chart
Optimizing	5.1 Organizational innovation and deployment		
	5.2 Causal analysis and resolution	planning game, peer review, the feedback during rapid cycles from the On-Site customer	Daily Scrums, Retrospectives

Table 2: Process enhancement model analysis contexts and patterns

Context	Pattern
Repeatable	2.1 Requirements management
	2.2 Project planning
	2.3 Project monitoring and control
	2.4 Supplier agreement management
	2.5 Measurement and analysis
	2.6 Process and product quality assurance
	2.7 Configuration management
Defined	3.1 Requirements development
	3.2 Technical solution
	3.3 Product integration
	3.4 Verification
	3.5 Validation
	3.6 Organizational process focus
	3.7 Organizational process definition
	3.8 Organizational training
	3.9 Integrated project management
	3.10 Risk management
Managed	4.1 Organizational process performance
	4.2 Quantitative project management
Optimizing	5.1 Organizational innovation and deployment
	5.2 Causal analysis and resolution

Furthermore, each pattern contains a set of problems, which represent the KPA practices. For each problem, there is a set of questions, which reflect actions, causes, and consequences to identify the actual problem (force) with that practice. For each problem, there is a solution, which is basically a set of agile practices. These sets of practices can help the company to address their improvement effort. In this regard, an example is provided in Table 3.

Figure 1 shows software process enhancement model analysing stage.

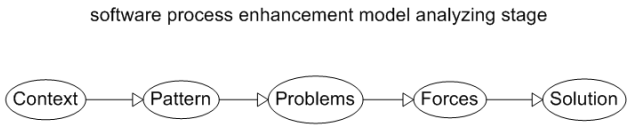

Fig 1 Process enhancement model analyzing stage

Based on the set of the solutions provided in the analysing staging, the gap between the current and the desired process will be identified and also all the stakeholders will be identified.

Table 3: An example of software process enhancement analysis

	Pattern	Problem	Force	Solution
Repeatable	2.1 Requirements management	Understand Requirements	Is there criteria for distinguishing appropriate requirements providers?	Scrum (Product Backlog) XP (user-stories)
			Is objective criteria for the evaluation and acceptance of requirements?	
			How to analyse requirements to ensure that established criteria are met?	
		Obtain Commitment to Requirements	How to negotiate and record commitments?	Scrum (Scrum board) XP (On-site customer)
			How to assess the impact of requirements on existing commitments?	

Afterwards, the change strategy, risks, budget, and resources will be determined in the planning stage. A request for sponsorship will be made, and if it is approved, go ahead will be given for the implementation, and if it is not approved, start over with the planning. Figure 2 shows software process enhancement planning stage.

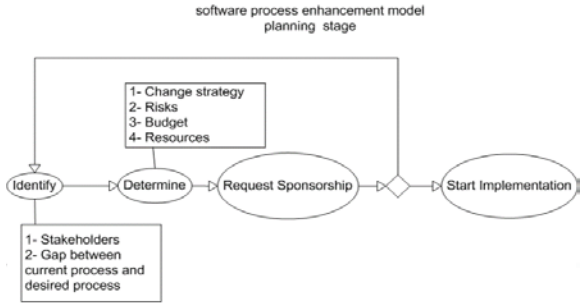

Fig 2 Software process enhancement model planning stage

After the request for sponsorship is approved, the implementing stage starts which determines the number of iterations and prioritize certain changes. Based on the number of iterations and prioritized changes, the process changes for this iteration is addressed. Furthermore, roles responsible for process changes are assigned and the infrastructure is prepared. Additionally, the training for the new process changes is given to the stakeholders. Figure 3 shows software process enhancement implementing stage. Finally, software process enhancement model represents a well-structured and systematic model to adopt the processes. However, these processes differ from one company to another. Therefore, this model's activities can be adjusted to suit the company processes and the adoption context because this model guides the processes changes at high-level abstraction.

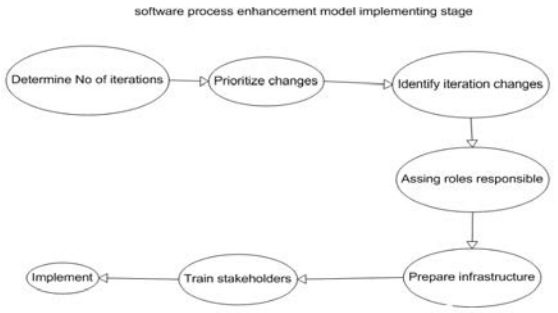

Fig 3 Software process enhancement model implementing stage

6. Validation

6.1 Determining critical success factors

Absolutely, it is important to determine the possible CSFs in order to validate the implementation of this model. The experts should do the validation based on high level of abstraction. The guideline for the right application of this model can be represented by CSFs, therefore, compliance can identify the model successful or not. For the determination of CSFs, the literature review was conducted to find out the most common CSFs. The following CSFs are the most common CFSs for any SPI model in light of previous studies [26, 27, 28].

1. Commitment
2. Staff Involvement
3. Training
4. Resources
5. Process Action Teams
6. Staff Experience
7. Guidance
8. Reviews - Feedback
9. Implementation Methodology
10. Monitoring
11. Communication
12. Return on Investment
13. Awareness of SPI

6.2 Focus group

Focus group discussion is about gathering a small group of experts to discuss a specific topic with the researcher. It usually is conducted in free-flowing style. This method is widely used in understanding experts' opinions on the software processes [29]. The primary purpose of the focus group discussion was to validate 13 CSFs found in the previous stage. In this regard, five experts were asked to join the group and all of them accepted the invitation. The sample consisted of two project managers and three senior software engineers. Their average age was 38 years. The group gathered on the site inside the company. The discussion lasted about 2 hours in October 2016. Each of the CSFs was discussed separately and at the end of the session, the validated CSFs were selected and put in a list, which the experts reviewed and gave the confirmation about the list. The experts eliminated three CSFs: return on Investment, implementation methodology and awareness of SPI. They also confirmed the following ten CSFs, which will be used as input for the third stage:

1. Commitment
2. Staff Involvement
3. Training
4. Resources
5. Process Action Teams
6. Staff Experience
7. Guidance
8. Reviews - Feedback
9. Monitoring
10. Communication

6.3 Delphi method

The Delphi method is usually a suitable method for researchers who lack data. Therefore, they need the opinion of experts. It is considered common and effective method in the researches on software process [30]. In this research, this method was used to find out the relative importance of the CSFs approved by the expert in the focus group discussion. There were send invitation emails to eighteen experts for the Delphi. Only ten agreed to participate; these experts were all men and their average age was 40 years. The literature suggests the experts for the Delphi panel should be between ten to eighteen people, the composition of the sample is ideal [31].The Delphi application had two rounds; the experts put CSFs in numerical sequence based on their importance in the first round, and they gave the definitive ranking in the second round.

In the first round, the list of the validated factors was sent by email to the experts and they were asked to give points to each factor based on its importance from 1 point to 10 points. The results of the first round are summarized in Table 4.

Table 4: Delphi first round summary

	Expert 1	Expert 2	Expert 3	Expert 4	Expert 5	Expert 6	Expert 7	Expert 8	Expert 9	Expert 10	Total
Factor 1	10	2	10	8	10	3	1	1	10	10	**65**
Factor 2	9	8	9	2	9	6	2	9	9	2	**65**
Factor 3	8	6	8	9	8	7	3	3	8	9	**62**
Factor 4	7	4	5	7	7	10	4	4	7	3	**58**
Factor 5	6	10	4	10	6	1	10	5	3	1	**56**
Factor 6	5	7	1	4	5	2	6	6	1	5	**42**
Factor 7	4	9	2	1	4	5	9	7	2	4	**47**
Factor 8	3	5	6	3	3	8	5	8	6	6	**53**
Factor 9	2	3	7	6	2	4	7	2	4	8	**52**
Factor 10	1	1	3	5	1	9	8	10	5	7	**50**
Total	**55**	**55**	**55**	**55**	**55**	**55**	**55**	**55**	**55**	**55**	**550**

In the second round, Table 4 prepared in the first round was shared with all the experts through a Skype group discussion, which lasted for 40 minutes. They thoroughly discussed the results especially factors 1 and 2 got more attention. All experts agreed to break the tie between factor 1 and 2 by voting. Out of 10, 7 experts voted in favor of factor 1; and 3 of them voted for factor 2; it means, factor 1 got extra 1 point to break the tie and became 66. In the end, even there were different opinions about the order of these factors, everybody accepted that the final results were fair. Table 5 shows the results.

7. Conclusion

In this paper, SPEM is presented that can help SMEs to enhance their software processes and address the effort for it. The model offers group of process patterns, which describes the typical problems for process improvement in SMEs, the patterns, are selected based on the existing weakness of the software process. These patterns are considered as reusable units that the company can adopt to achieve the desired CMMI maturity level. The model contains three stages: analysing, planning and implementing. The CSFs to validate the implementation of this model are determined using focus group method and their importance are addressed using Delphi method. The future steps to evaluate the model based these CSFs in industrial context. Furthermore, the model can be extended to include the other SPI aspects like the organizational aspects and managerial aspects using the same core idea of SPEM by adding new patterns to cover these aspects.

Table 5: Delphi final result

	Factor No	Factor name	Factor points
Position 1	1	Commitment	**65+1**
Position 2	2	Staff involvement	**65**
Position 3	3	Training	**62**
Position 4	4	Resource	**58**
Position 5	5	Process action team	**56**
Position 6	8	Reviews – feedback	**53**
Position 7	9	Monitoring	**52**
Position 8	10	Communication	**50**
Position 9	7	Guidance	**47**
Position 10	6	Staff experience	**42**

References

[1] A. M. AL-Ashmori, B. Bashiri Rad, and I. Suhaimi, "Software process improvement frameworks as alternative of CMMI for SMEs: a literature review". Journal of SE, vol. 11, no. 2, pp, 123-133, March 2017.

[2] F. J. Pino, F. García, and M. Piattini M, "Software process improvement in small and medium software enterprises: a systematic review". SQ Journal, vol.16, no. 2, pp. 237-261, June 2008.

[3] I. Garcia, and C. Pacheco, "Toward automated support for software process improvement initiatives in small and medium size enterprises," in Software engineering research, management and applications, ed, R. Lee, and N. Ishii, pp 51-58, Springer, Berlin, 2009.

[4] P. V. Martins, and A. R. da Silva, "PIT-ProcessM: A Software Process Improvement Meta-model, " IEEE 7th International Conf. on the Quality of Information and Communications Technology (QUATIC), pp. 453-458, Sept. 2010.

[5] I. Sommerville, Software Engineering, 9th ed. Pearson, New York, 2011.

[6] A. M. Solyman, O. A. Ibrahim, and A. A. Elhag, "Project management and software quality control method for small and medium enterprise," IEEE International Conf. on InComputing, Control, Networking, Electronics and Embedded Systems Engineering (ICCNEEE), pp. 123-128, Sept. 2015.

[7] M. A. T. Almomani, S. Basri, A. K. B. Mahmood, and A. O. Bajeh, "Software development practices and problems in Malaysian small and medium software enterprises: a pilot study," 5th International Conf. on IT Convergence and Security (ICITCS), IEEE. Kuala Lumpur, Malaysia, pp. 1-5, Aug. 2015.

[8] F. McCaffery, M. Pikkarainen, and I. Richardson, "Ahaa --agile, hybrid assessment method for automotive, safety critical SMEs," ACM/IEEE 30th International Conf. on Software Engineering, pp. 551-560, May 2008.

[9] D. M. Ahern, A. Clouse, and R. Turner, CMMI Distilled: A Practical Introduction to Integrated Process Improvement, Addison-Wesley, Boston, 2004.
[10] D. Galin, and M. Avrahami, "Are CMM Program Investments Beneficial? Analyzing Past Studies," IEEE Software, vol. 23, no. 6, pp. 81-87, Nov. 2006.
[11] P. Clarke, and R. O'Connor, "Towards the identification of the influence of SPI on the successful evolution of software SMEs," Proc. 18th International Conference on SQM, pp. 29- 40, 2010.
[12] K. C. Dangle, P. Larsen, M. Shaw, and M. V. Zelkowitz, "Software process improvement in small organizations: a case study," IEEE Software, vol. 22, no.6, pp. 68-75, Nov. 2005.
[13] M. Habib, S. Ahmed, A. A. Rehmat, M. J. Khan, and S. Shamail, "Blending six sigma and CMMI-an approach to accelerate process improvement in SMEs," IEEE International Multitopic Conf., pp. 386-391, Dec 2008.
[14] J. Iqbal, R. B. Ahmad, M. H. Nasir, M. M. Niazi, S. Shamshirband, and M. A. Noor, "Software SMEs' unofficial readiness for CMMI®-based software process improvement," SQ Journal, vol.24, no.4, pp. 997-1023, Dec. 2016.
[15] N. Nikitina, and M. Kajko-Mattsson, "Guiding the adoption of software development methods," Proc. International Conf. on Software and System Process, ACM, Nanjing, China, pp. 109-118, May 2014.
[16] CROSSTALK. "Beyond the Agile Manifesto. CrossTalk," Journal of DSE, vol. 29, no. 6, pp. 40-41, Nov. 2016.
[17] M. I. Khan, M. A. Qureshi, and Q. Abbas, "Agile methodology in software development (SMEs) of Pakistan software industry for successful software projects (CMM framework), " IEEE International Conf. on Educational and Network Technology, pp. 576-580, June 2010
[18] K. Łukasiewicz, and J. Miler, "Improving agility and discipline of software development with the Scrum and CMMI," IET Software, vol.6, no. 5, pp. 416-422, Oct. 2012
[19] Z. Lina, and S. Dan, "Research on Combining Scrum with CMMI in Small and Medium Organizations," International Conf. on Computer Science and Electronics Engineering (ICCSEE), pp. 554-557, March 2012.
[20] M. A. Muñoz-Mata, J. Mejia-Miranda, and C. Valtierra-Alvarado, "Helping organizations to address their effort toward the implementation of improvements in their software process," Revista Facultad de Ingeniería Universidad de Antioquia, no 77, pp. 115-126, Dec. 2015.
[21] M. Saunders, and P. Tosey, The layers of research design. Rapport, Winter, pp. 58-59, 2012
[22] D. Della Porta, and M. Keating, Approaches and methodologies in the social sciences: A pluralist perspective. Cambridge University Press, New York, 2008.
[23] E. Herranz, R. Colomo-Palacios, and A. de Amescua Seco, "Gamiware: a gamification platform for software process improvement," European Conf. on Software Process Improvement. Springer, pp. 127-139, Sept. 2015.
[24] J. W. Creswell, 4th ed., Research design: Qualitative, quantitative, and mixed methods approaches, Sage publications, USA, 2013.
[25] A. M. AL-Ashmori, B. Bashiri Rad, A. S. S. AL-Ameri, and Ahanin, Z, "Evaluating the readiness to adopt CMMI in Malaysian software SME," Indian Journal of ST, vol.9, no. 48, 2016.
[26] Z. Habib, The critical success factors in implementation of software process improvement efforts: CSFs, motivators & obstacles. University of Gothenburg Department of Applied Information Technology Gothenburg, Sweden, 2009.
[27] E. Kouzari, V. C. Gerogiannis, I. Stamelos, and G. Kakarontzas, "Critical success factors and barriers for lightweight software process improvement in agile development: A literature review," 10th International Joint Conf. on Software Technologies (ICSOFT), vol. 1, pp. 1-9, July 2015.
[28] G. K. Viju, M. M. Elsalam, K. A. Ibrahim, and M. Jassim, "The Impact of Software Process Improvements in Small and Medium Scale Enterprises". International Journal of Soft Computing and Engineering (USCE), vol. 3, no. 4, Sept. 2013.
[29] H. B. Christensen, and K. M. Hansen, "An empirical investigation of architectural prototyping," Journal of SS, vol.83,no.1, pp. 133-142, Jan 2010.
[30] J. Iden, and L. Langeland, "Setting the stage for a successful ITIL adoption: A Delphi study of IT experts in the Norwegian armed forces," Information systems management, vol. 27, no. 2, pp. 103-112, Apr. 2010.
[31] R. Colomo-Palacios, P. Soto-Acosta, F. J. García-Peñalvo, and A. García-Crespo, "A study of the impact of global software development in packaged software release planning," Journal of UCS, vol. 18, no. 19, pp. 2646-2668, Nov 2012.

Ammar Mutahar AL-Ashmori received his M.Sc. of Software Engineering in 2017 from Asia Pacific University of Technology and Innovation (APU), Kuala Lumpur Malaysia. His research interest focuses some different areas in software engineering and information technology management including software process improvement, software quality engineering, business modelling, organizational cultures, and enterprise.

Dr Babak Bashari Rad received his M.Sc. of Computer Engineering (Artificial Intelligence and Robotics) in 2001 from University of Shiraz; and Ph.D. of Computer Science in 2013 from University technology of Malaysia. Currently, he is the program leader of postgraduate studies and senior lecturer in school of computing, Asia Pacific University of Technology and Innovation (APU), Kuala Lumpur Malaysia. His main research interest covers various areas in computer science and information technology including Information Security, Malware Detection, Machine Learning, Artificial Intelligence, Image Processing, Robotics, and other relevant fields.

Zahra Ahanin received her M.Sc. of Software Engineering in 2015 from Asia Pacific University of Technology and Innovation (APU), Kuala Lumpur Malaysia. Currently, she is doing her Ph.D. of Information Systems in University of Malaya. Her research interest covers some different areas in Software Engineering and Information Systems including System Development, Data Mining, Machine Learning, Opinion Mining and other relevant fields.