Critical parameters for the adoption of software capability maturity model by small and medium enterprises

Christine Amulen, Geoffrey Mayoka G. Kituyi* and Edward Kabaale

Department of Business Computing,
Makerere University Business School,
Uganda
Email: camulen@mubs.ac.ug
Email: kimayoka@gmail.com
Email: ekabaale@mubs.ac.ug
*Corresponding author

Abstract: The study involved a survey of Ugandan small and medium enterprises that develop software on critical factors influencing the adoption of software capability maturity model. The research examined the software development environment in these small and medium enterprises and the operational tenets of the software capability maturity modelling in order to establish the critical parameters that should be considered for successful adoption of the software capability maturity model by small and medium software enterprises. Quantitative data was analysed using factor analysis to deduce the critical factors on the variables based on the data collected from the field. Linear regression analysis was also done to determine whether there was a significant relationship between adoption of software capability maturity model and adoption parameters. The key parameters for adoption of the software capability maturity model were identified as management's ability to choose appropriate improvement strategy, organisational culture, managements' commitment, developers' involvement, effective communication, rewards, training and project championship.

Keywords: software capability maturity model; SW-CMM; critical parameters; adoption; small and medium enterprises; SMEs.

Reference to this paper should be made as follows: Amulen, C., Kituyi, G.M.G. and Kabaale, E. (2016) 'Critical parameters for the adoption of software capability maturity model by small and medium enterprises', *Int. J. Agile Systems and Management*, Vol. 9, No. 1, pp.1–20.

Biographical notes: Christine Amulen is a Lecturer in the Department of Business Computing at Makerere University Business School. She received her BBA and MSc in Computer Science from Makerere University. Her research interest is in the areas of agile computing, software quality and software development.

Geoffrey Mayoka G. Kituyi is a Senior Lecturer in the Department of Business Computing, Makerere University Business School where he has taught since 2006. He is also an Assistant Professor of ICT at The ICT University, Cameroon. He has over 40 peer-reviewed journal publications in the area of information systems and technology. He has also published two textbooks on

database programming and business computing. His research interest is in the area of information systems design for sustainability in resource constrained environments. He received his Bachelor of Business Computing, PPD ICT Policy and Regulation, and MSc in Information Systems, all from Makerere University, and is currently pursuing his PhD in ICT at the ICT University.

Edward Kabaale is a Lecturer in the Department of Business Computing, Makerere University Business School where he has taught since 2006. He has over ten peer-reviewed journal publications in the area of information technology. His research interest is in the area of artificial intelligence. He received his Bachelor of Business Computing and his MSc in Information Systems from Makerere University. He is currently pursuing his PhD in Information Systems at Griffin University, Australia.

This paper is a revised and expanded version of a paper entitled 'A software capability maturity adoption model for small and medium enterprises' published by Electronic Journal of Information Systems in Developing Countries (2012) Vol. 55, Issue. 1, pp.1–19.

1 Introduction

Software process improvement (SPI) is a current topic in the software engineering community and has received much attention in both academia and industry. SPI's major aim is to improve the effectiveness of the software development processes. In order to maintain a competitive edge, most organisations plan to apply SPI with the desired results of predictability of output in the software development process, considering time, cost and quality. Paulk et al. (1995) argue that SPI can also focus on improving specific working practices in an organisation.

Several approaches have been developed for SPI, including the Software Engineering Institute's software capability maturity model (SW-CMM), capability maturity model integration (CMMI) and ISO's software process improvement capability determination (SPICE). Hence, SPI can be based on a road map like the SW-CMM, SPICE, Bootstrap, ISO9000 and other agile methodologies.

Batista and Dias de Figueiredo (2000) argue that the SW-CMM helps organisations to provide the infrastructure for achieving a disciplined and mature quality software process. Efforts put into the above approaches can assist software developing organisations in producing high quality software. The SW-CMM has become quite popular as a model for process improvement for software developing organisations, especially in the USA and also recently in Europe (Madachy, 2007; Paulk, 2001). A number of large organisations in developed countries have used the SW-CMM and published its benefits. However, there has been almost attention to SW-CMM in developing countries. Even within developed countries, it is only large software companies using the model (Delroy and Duggan, 2007). This is the same case in Uganda, where most software companies are small. Moreover, the software development industry is increasingly becoming active in developing world (Delroy and Duggan, 2007). Hence for small organisations to benefit from these new opportunities and become better competitors in the software market, they must adopt and demonstrate internationally accepted software process practices.

2 Challenges to adoption of SW-CMM in small and medium enterprises

The challenge of software adoption in small and medium enterprises (SMEs) lies in the constraints of small settings, which are often limited in the people, resources and skills that can be called on. Dangle et al. (2005) established that many organisations now understand the importance of focusing on the quality of software process. Yet ability to produce reliable and usable software within time and budget is still difficult to achieve for many organisations. Many of these organisations now search for solutions from SPI. Further, the value for the customer has become a very important driving force for many software developing organisations. It is of vital importance for survival in the ever increasing competition. This calls for initiatives for improvement efforts.

The past decade has seen many studies SPI zeroing on traditional software development approaches, examining their strengths and weaknesses (Jung and Goldenson, 2009). Today, however, several developments have culminated into the adoption of agile software development methodologies, applied concurrently with traditional methodologies. Hence, several challenges of adoption have emerged. For example, while traditional approaches bring in leverage in terms of application, Abrahamsson and Salo (2007) observe that traditional SPI approaches do not provide for the much needed interactivity during SPI. Moreover, agile methodologies for SPI have been found to significantly minimise the time spent on SPI, product risks and costs. This is in addition to ensuring good quality and more reusable software (Trendowicz et al., 2009). However, most of the SPI initiatives in small and medium software organisations fail to materialise (Dyba and Dingsoyr, 2008). Dubé and Robey (1999) observe that two out of three SPI initiatives fail. This study therefore sought to examine the critical parameters for adopting of SW-CMM in small and medium software developing enterprises.

According to Johnson and Brodman (1997), small organisations are those that employ less than 50 people, and small projects are those with less than 20. These definitions are not universal. For example, in the USA, small enterprises are those employing less than 500 people. On the other hand, the SMEs in Mexico refer to organisations employing less than 100 people. In Ireland, small organisations are those employing 50 or less people. In Uganda, a small enterprise employs 1 to 50 people, while a medium enterprise employs 50 to 100 (Kasekende and Opondo, 2003). This study adopted Kasekende and Opondo's definition of SME and applied it in the selection of participants.

3 Research design

A quantitative research design was used, whereby reported findings are quantitative in nature. Quantitative research methods included design of survey questionnaire and descriptive analysis techniques.

3.1 Study population

The population of the study constituted mainly of 30 small and medium software developing enterprises in and around Kampala city, Uganda. The study sample size was influenced by the time available, the number of organisations involved in software

development and necessary degree of precision. The survey questionnaire was given mainly to the key people who were actively involved in developing software using purposive sampling techniques. This was necessitated by the need to involve only participants with some level of knowledge of SPI and SW-CMM.

3.2 Data collection and research instruments

A self-administered questionnaire was designed and used as the major instrument for collection of primary data. The choice of self administered questionnaire was based on the fact that it was easy to administer given the high number of participants. It is also reported that self-administered questionnaires are most suitable for computer-based research methods (Fink, 2009; Klein and Myers, 2001). The questions were designed to incorporate all the issues entailed in the variables of study.

3.3 Quality of research tools

According to Klein and Myers (2001), reliability of a study instrument is an important aspect that must be put into consideration by the researcher so as to obtain correct information. The questionnaire was carefully designed so as to yield valid information. Particular attention was paid to ensure that the questions were relevant, appropriate, precise and unbiased as stipulated by Fink (2009) and Jarvinen (2000).

In addition to the above measure, the study questionnaire was tested and piloted to establish its reliability. Reliability of the instrument was measured using Cronbach's alpha coefficient. The reliability analysis results for all variables were above 0.5.

3.4 Data processing and analysis

Data analysis was done using statistical software (SPSS) version 12. Descriptive statistics were done and results presented in pie charts and bar charts for easy interpretation. Factor analysis using principal component analysis method was used to deduce the critical factors on the variables based on the data collected from the field.

Linear regression analysis was also done to determine whether there was a significant relationship between the variables i.e. dependent (adoption of SW-CMM) and the independent (factors/parameters for adoption).

4 Findings

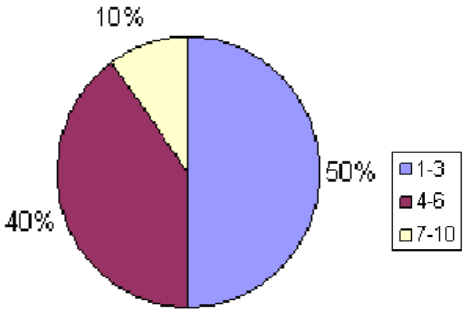
4.1 Duration in software development

An analysis was done to determine the duration that different software organisations had been in software development business. This was aimed at establishing their experience in software development. Figure 1 presents the duration in years that the different software developing organisations had been in software development business.

The pie chart as seen in Figure 1 depicts that only 10% of the companies surveyed had been in software development for over six years. This gives the impression that software development is an emerging business in most small and medium organisations in Uganda.

On the other hand, 40% of the organisations had been in practice for over four years. These might be an indication that the business is growing but growing steadily as more people are joining the industry. However, 50% of the companies had just joined the industry.

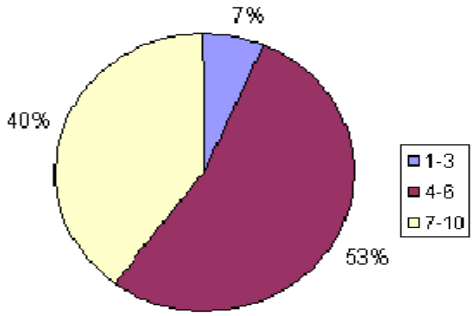
Figure 1 Duration in software development in years (see online version for colours)



4.2 Employees in active software development

Further, descriptive analysis was carried out to establish the characteristics of respondents, like establishing how long they had been in software development practices and also to find out those respondents who were actively involved in software development in small software enterprises. Figure 2 shows the number of respondents who were actively involved in software development in small enterprises.

Figure 2 Employees in active software development (see online version for colours)



The findings as seen in the Figure 2 reveal that 53% of the companies employ 4 to 6 people in software development. While 40% employ 7 to 10 people and 7% employ 1 to 3 people. This gives an impression that software development enterprises in Uganda employ between 4–10 people due to their small scale production. This could also be a strategy of minimising costs which has always been the characteristic of small enterprises. From the above, it is also observed that all companies that participated in the study qualify to be called small enterprise in line with the definition of Kasekende and Opondo (2003) that small organisations employed less than 50 people and Johnson and Brodman (1997) that small projects had less than 20 workers.

4.3 Key roles in software developing SMEs

Analysis was done to determine some of the roles found in software developing enterprises surveyed as seen in Figure 3:

Figure 3 Key roles in software developing SMEs (see online version for colours)

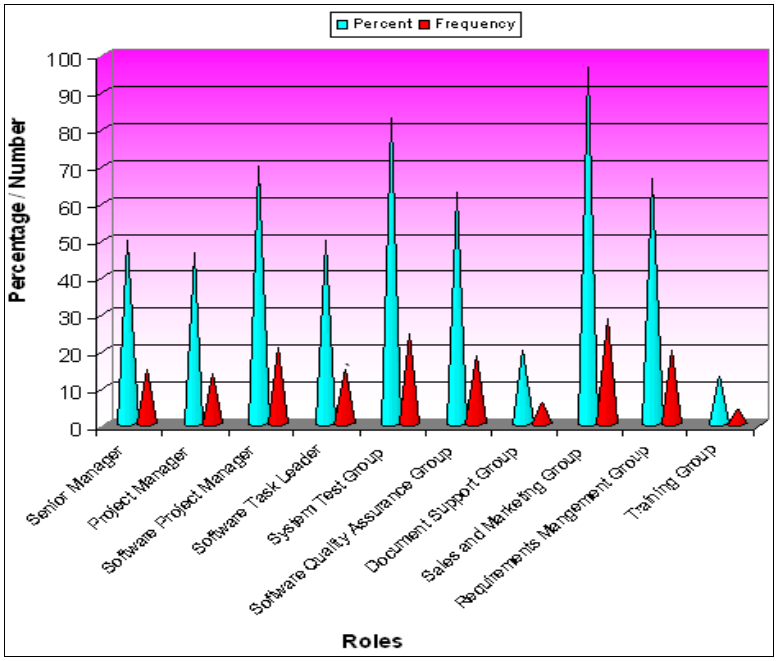


Figure 3 shows the major roles found in small software developing enterprises/organisations in Uganda. Using factor analysis, ten roles were extracted as the key roles found in small software developing enterprises in Uganda as compared to the original 25 roles stated in the original SW-CMM. The roles include

1 senior manager
2 project manager
3 software project manager
4 software task leader
5 system test group
6 software quality assurance group
7 document support group
8 sales and marketing group
9 requirements management group
10 training group.

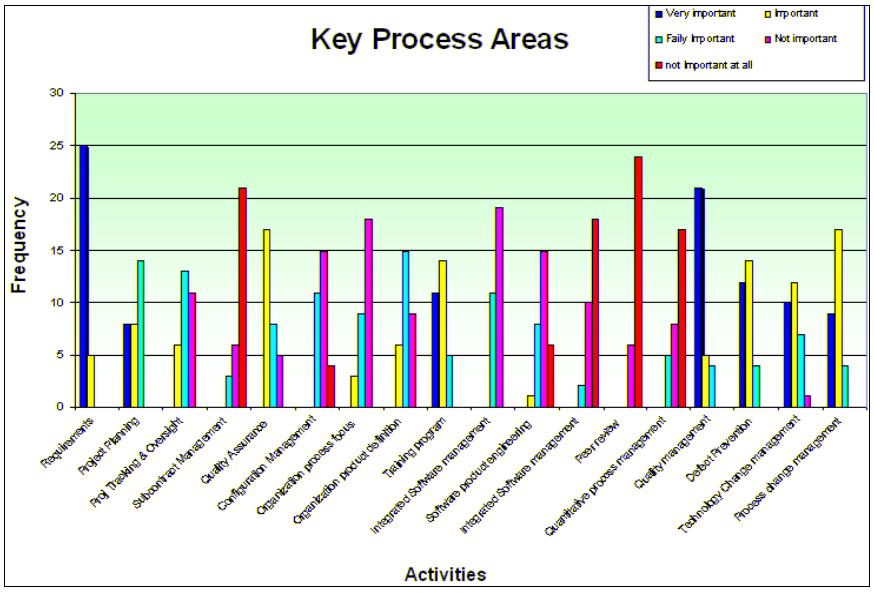
This finding implies that small software developing companies have fewer roles as compared to the 25 roles spelt out in the original SW-CMM.

Having all the 25 roles in a small organisation where usually the prime goal is to minimise costs seems to be unreasonable. An obvious argument is that, there is no consistency between the number of roles, responsibilities and tasks to be done. With the number of staff available in a small software organisation, the above identified ten roles would therefore be realistic in a small setting.

Another reason is that most of the time there are not even enough people to fill these roles and take part in groups in small software enterprises. This is due to the fact that they employ between six to ten people. Therefore, considering the success of SW-CMM in many organisations around the world and its focus on continuous process improvement, the SW-CMM roles can be scaled down to at least ten as indicated in these findings.

It should also be noted that, the roles identified are not found in one organisation, but they are the ones identified as key in different small organisations surveyed. This is because in small companies one individual can handle different roles or tasks. This observation is in line with scholars who argue that there is a high probability that some of the roles and groups defined in the SW-CMM may not be present in a small software organisation. Therefore, some of these roles can be naturally omitted. For example, if a small organisation does not subcontract, it is pointless to have such a role in the organisation.

Figure 4 Key software practices in SMEs (see online version for colours)



4.4 Key software practices in small software developing enterprises

Analysis was done to determine the major activities practiced in small software enterprises. Findings reveal that requirements management, quality management and training were ranked as very important activities or practices, while activities like

subcontract management and peer reviews were ranked as not important at all. Findings are presented in Figure 4. These were the basis for the modified SW-CMM (see Kituyi and Amulen, 2012).

4.5 Critical parameters for successful adoption of the SW-CMM

The main objective of the study was to establish the critical parameters and/or factors for successful adoption of the SW-CMM. This was the independent variable, while adoption of SW-CMM was independent variable. As described in the methodology, data was analysed using factor analysis to extract the key parameters that should be considered for successful adoption of the SW-CMM in small software developing enterprises. Table 1 presents factor analysis results.

Table 1 Factor analysis of variables

Factor analysis		
Communalities		
	Initial	*Extraction*
D1	1.000	.822
D2	1.000	.610
D3	1.000	.632
D4	1.000	.655
D5	1.000	.588
D6	1.000	.754
D7	1.000	.766
D8	1.000	.728
D9	1.000	.849
D10	1.000	.716
D11	1.000	.662
D12	1.000	.739
D13	1.000	.832
D14	1.000	.764
D15	1.000	.668
D16	1.000	.810
D17	1.000	.569
D18	1.000	.847
D19	1.000	.715
D20	1.000	.694
D21	1.000	.774

Note: Extraction method: principal component analysis

Table 1 depicts the variables that were analysed in the study. These 21 factors were thought to influence the adoption of the SPI in small software developing enterprises.

The variables were coded as represented by D1–D21. These represent; provision of adequate resources, managements support, managements active participation, emphasis on process improvement, training all technical workgroups, training software developers, adoption of new technologies, improving employee competence, developers initiating changes to the existing process, individuals to coordinate the process, identifying the projects, methods of communication, having champions, written organisation procedures, clear policies on SPI, rewards, identifying core strategies, employee involvement in decision making, choice of appropriate strategy and the involvement of all developers in SPI initiatives respectively.

Using principal component analysis method, only those factors that had a loading greater than 0.5% were considered significant. Out of the 21 factors, eight were identified as key parameters for adoption of SW-CMM. The proportion of the variance explained by the eight parameters was computed to be 72.3%. Table 2 presents the principal component analysis results:

Table 2 Principal component analysis results

Total variance explained									
Components	*Initial eigenvalues*			*Extraction sums of squared loading*			*Rotation sums of squared loading*		
	Total	*% of variance*	*Cumulative %*	*Total*	*% of variance*	*Cumulative %*	*Total*	*% of variance*	*Cumulative %*
1	3.071	14.622	14.622	3.071	14.622	14.622	2.311	11.005	11.005
2	2.791	13.290	27.911	2.791	13.290	38.551	2.092	9.960	21.055
3	2.234	10.640	38.551	2.234	10.640	38.551	2.092	9.960	31.015
4	1.874	8.926	47.477	1.874	8.926	47.477	2.037	9.100	40.714
5	1.503	7.156	54.633	1.503	7.156	54.633	1.818	8.658	49.373
6	1.464	6.973	64.605	1.464	6.973	64.605	1.721	8.194	57.567
7	1.233	5.869	67.475	1.233	5.869	67.475	1.618	7.705	65.272
8	1.023	4.872	72.346	1.023	4.872	72.346	1.486	7.074	72.346
9	.994	4.733	77.079						
10	.806	3.836	80.916						
11	.738	3.514	84.430						
12	.700	3.335	87.764						
13	.630	2.998	90.762						
14	.475	2.261	93.024						
15	.439	2.093	95.116						
16	.310	1.474	96.591						
17	.230	1.094	97.685						
18	.204	.969	98.654						
19	.143	.681	99.336						
20	.105	.499	99.835						
21	.035	.165	100.00						

Note: Extraction method: principal component analysis

Further to the above analysis, rotated component matrix Table 3 was used to identify and group the major eight factors that influence the adoption of the software process improvement strategy (SW-CMM) in small software developing enterprises. From the table above, only variables with factor loadings greater than 0.5 in absolute values considered significant. Table 3 presents rotated components.

Table 3 Rotated component matrix

Rotated component matrix[a]								
	Component							
	1	*2*	*3*	*4*	*5*	*6*	*7*	*8*
D1	.763	–.077	–.122	.145	–.209	.377	–.062	.095
D2	–.139	–.295	–.108	.000	–.315	.082	.463	–.413
D3	.543	.066	–.247	–.059	.465	–.118	–.191	–.032
D4	–.087	.803	.018	–.010	–.028	.019	–.011	.020
D5	.101	.286	–.334	–.330	–.115	.469	.158	.129
D6	.171	.023	.151	–.152	–.035	.099	.816	–.037
D7	.016	–.351	–.032	–.015	–.718	–.124	.330	.036
D8	.033	.430	.105	.409	.005	–.526	.296	.011
D9	–.196	.481	.691	.198	.146	.104	.120	.124
D10	–.815	.116	.071	.047	.005	.048	–.167	.024
D11	.283	–.117	.146	.102	.262	–.576	–.164	.330
D12	.059	.032	.066	.845	–.055	.024	–.088	.073
D13	.129	–.180	–.163	.471	–.015	.108	.215	.690
D14	.120	–.148	–.024	.230	–.011	.141	.188	–.787
D15	–.607	–.148	.133	.345	.121	.333	–.111	.042
D16	–.160	–.201	–.055	–.100	.844	.029	.100	.084
D17	.056	–.022	.204	.029	.215	.684	.088	–.036
D18	.215	.310	–.814	.079	.016	.047	–.176	.060
D19	.312	.079	.102	–.693	.080	.269	.041	.200
D20	.087	.587	.166	–.234	.129	–.045	–.485	.073
D21	–.032	.394	.747	–.034	–.190	.052	–.138	–.032

Notes: Extraction method: principal component analysis
Rotation method: variamax with Kaiser normalisation
[a]Rotation converged in 41 iterations.

Table 4 presents the modal summary generated based on the dependent variable that was adoption of the SW-CMM and the independent variable which was the parameters necessary for the adoption of the SW-CMM. The model predicts 46.4% of the variance in the dependent variable (adoption of SW-CMM).

Table 4 Model summary

Model summary					
Model	*R*	*R square*	*Adjusted R square*	*Std. error of the estimate*	
1	.681[a]	.464	.086	2.12088	
ANOVA					
Model		*Sum of squares*	*df*	*Mean square*	*F*	*Sig.*
1	Regression	66.232	12	5.519	1.227	.341[a]
	Residual	76.468	17	4.498		
	Total	142.700	29			

Note: [a]Pearson R is closer to one, meaning strong relationship.

Table 5 summarises a regression of the variables. It shows that only variables with significance of less than 0.05 or 5% were identified as significant at 95% confidence level. These were the parameters considered to be key parameters for the adoption of SW-CMM.

Table 5 Regression analysis of the variables

Coefficients						
Model		*Unstandardised coefficients*		*Standardised coefficients*		
		B	*Std. error*	*Beta*	*t*	*Sig.*
1	(Constant)	53.900	.387		139.198	.000
	REGR factor score 1 for analysis 1	.143	.543	.065	.264	.795
	REGR factor score 2 for analysis 1	–1.262	.453	–.569	–2.784	.013
	REGR factor score 3 for analysis 1	–.037	.468	–.017	–.079	.938
	REGR factor score 4 for analysis 1	–.229	.435	–.103	–.526	.605
	REGR factor score 1 for analysis 2	–.081	.424	–.037	–.192	.850
	REGR factor score 2 for analysis 2	–.882	.436	–.398	–2.025	.059
	REGR factor score 3 for analysis 2	–436	.453	–.197	–.963	.349
	REGR factor score 4 for analysis 2	.848	.480	.382	1.766	.095
	REGR factor score 5 for analysis 2	–.299	.442	–.135	–.676	.508
	REGR factor score 6 for analysis 2	.600	.412	.271	1.458	.163
	REGR factor score 7 for analysis 2	–.084	.410	–.038	–.204	.841
	REGR factor score 8 for analysis 2	.144	.436	.051	.260	.798

5 Discussion of findings

The survey examined 21 variables out of which only eight were identified using factor analysis as critical parameters for the adoption of SW-CMM in small software

companies. The eight important factor for the adoption of SW-CMM for SPI in small software companies are:

1 management commitment

2 management ability to choose appropriate improvement strategy

3 developer's involvement

4 effective communication

5 organisations' culture

6 having reward systems

7 training

8 having people to champion SPI.

The two most critical factors are:

1 managements' commitment and developers' involvement

2 management's ability to choose appropriate improvement strategy.

These finding agree with studies of Kan et al. (1994) who argue that management commitment is important for quality software. The findings also are consistent with Fitzpatrick (1996) on the issue of management ability to choose appropriate improvement strategies and Kituyi and Amulen (2012) on developer involvement in the SPI activities. Issues effective communication, organisational culture (change management), reward systems, training and championship have all been suggested by various researchers as important factors for improved software and or information systems (e.g., see, Kituyi et al. 2012; Kituyi and Tusubira, 2013; Mellon, 2007; Paulk et al., 1995).

6 Conclusions and recommendations

The results obtained in this study will help small and medium software companies improve on software quality by adopting the SW-CMM. The original SW-CMM had 21 factors. This made it difficult and complex for small software companies to implement. This study proposes only eighth important factors for consideration in the adoption and use of SW-CMM by small and medium software companies. Hence it will be less complex for small companies to improve software quality following the condensed SW-CMM. The proposed factors are explained in details as follows:

6.1 Management's ability to choose appropriate improvement strategy

Management's ability to determine the appropriate SPI strategy was identified as the most critical parameter/step of the process of founding a mature organisation. Management must be able to choose an appropriate model. Management must be able to choose an appropriate model. The different sizes of companies can be divided according to the number of employees and the number of products under development.

The current problem with SPI is however said to be not lack of standards or models, but rather lack of an effective strategy to successfully implement these standards or

models. Therefore, an organisation should be able to determine the appropriate SPI strategy.

The importance of SPI implementation demands that it should be recognised as a complex process in its own right (Ravi and Ruppel, 2010; Richardson, 2001). Organisations should then determine their SPI implementation maturity through an organised set of activities that are considered vital and achievable.

6.2 Organisational culture

Organisational culture was identified as the second most critical parameter to consider when adopting SW-CMM in SPI. When proposing, implementing and deploying process improvements strategies, it is very important to put into consideration the organisational cultures. Small software enterprises surveyed believed that, despite of the benefits of the SW-CMM, it was not designed for them but for large software houses. This cultural belief eventually creates an impact on the adoption of such models (Jones, 2005).

Process improvement affects more than just the processes used by practitioners to perform their work. Process change means cultural change, replete with all the difficulties inherent in changing the perceptions, values, and normative behaviours of a community. Some of the forces that make such improvement efforts difficult are:

- Resistance to change often due to perceived threat of losing power, control, familiarity, or social status also was identified as a key factor (Calvo-Manzan et al., 2012).
- The existing tolerance and readiness for change present within the current organisational climate. Process change imposes a learning curve, which typically makes things appear to get worse before they get better. Improvement efforts consume time and resources, which many prefer to, spend on their particular development projects of organisational change.
- It is therefore imperative to take into consideration individual organisations cultures and try not to adopt solutions considered to be against these cultures. Otherwise, either the initiatives will not be adopted or they will be adopted in an inefficient way, thus affecting process compliance and performance.

The following factors were considered important though not critical in the process of adoption of any SPI strategy.

6.3 Managements' commitment

Senior managers must absolutely be convinced about the initiative's relevance from the beginning. Managements' commitment and support helps throughout the project by providing resources. This is in line with Hedelin and Allwood (2002), that to be successful, every activity in an organisation must have a sponsor with budgeting authority. A senior manager at the highest possible level must be convinced and motivated to ensure the initiatives succeed. Organisations trying to perform non-sponsored initiatives often do not succeed (Muller et al., 2009; Dubé and Robey, 1999).

6.4 Developers' involvement

Developers' involvement is another key factor to consider in proposing and designing improvements. According to Ian (1992), this helps to institutionalise the improvements quickly and smoothly. Obtaining this degree of participation is however not easy with larger groups. Hence, larger organisations might limit developers' participation, involving only a few of them in the initiative. But the size of small enterprises can help them actively involve everyone in SPI.

6.5 Effective communication

Effective communication was also identified as a key factor for the adoption of SW-CMM in small software companies. This observation is in line with Abrahamsson and Salo (2007) that communication is one of the most effective tools an organisation can use to obtain acceptance of change.

It is believed that the size of organisations can facilitate effective communication. Generally, communication is more effective in small groups. Hence, small software companies should easy attain effective communication given that they normally have smaller project teams. Such communication should be face-to-face since passive participation as in written memos typically does not demonstrate the necessary commitment. Communication must also occur frequently and an organisation should encourage its executives and middle-level managers to initiate it.

6.6 Rewards

Taking an individualistic reward approach such as giving incentives to individuals for performing process improvement related activities can play a big role in encouraging staff to take on active involvement on SPI. Pankaj (2001) argues that that this enables staff to propose and implement improvements on an individual basis. This makes them view either success or failure in the SPI initiative as success or failure of the whole software development unit.

6.7 Training

The findings also showed that for any new project to succeed there is need to undertake process related training. Persse and Wiley (2001) argue that all developers in the software development unit need to be trained on all the processes and roles in the SW-CMM. For example, the developers, quality assurance people and configuration managers should undergo the full project-manager training course. Given the size of the enterprises, it is possible to train all the technical workgroups on the processes being used. This, according to Pankaj (2002) will help the developers better understand what, how and why they were developing software in a certain way. However, given the financial strain small enterprises experience, coupled with their objective of minimising costs while maximisation profits, it might become an expensive venture. To make it more affordable, training can be reduced to training individual persons in the specific where they must execute, without giving each one a detailed training on others roles.

6.8 Champions

Bollinger et al. (2009) argue that an effective way to convince everyone of the importance of improving the software process is to have an SPI champion on the software development team. These are the persons responsible for spreading the process improvement vision and convince developers to participate actively in the initiative. These champions however, cannot be designated but may arise naturally from the group.

References

Abrahamsson, P. and Salo, O. (2007) 'An iterative improvement process for agile software development', *Software Process Improvement and Practice*, Vol. 12, No. 1, pp.81–100.

Batista, J. and Dias De Figueiredo, A. (2000) 'Software process improvement in a very small team: a case of capability maturity model', *Journal of Software Maintenance and Evaluation*, Vol. 5, No. 4, pp.243–250.

Bollinger, T., Clement, L. and McGowan (2009) 'A critical look at software capability evaluations', *IEEE Software*, Vol. 26, No. 5, pp.80–83

Calvo-Manzan, J.A., Gonzalo, C., Gerzón, G., Jezreel, M., Mirna, M. and Tomás, S.F. (2012) 'Methodology for process improvement through basic components and focusing on the resistance of change', *Journal of Software Evolution and Process*, Vol. 24, No. 5, pp.511–523.

Dangle, K.C., Larsen, P., Shaw, M. and Zelkowitz, M.V. (2005) 'Software process improvement in small organizations: a case study', *Software*, Vol. 22, No. 6 pp.68–75, IEEE.

Delroy, C. and Duggan, E.W. (2007) 'Modified capability framework for improving software processes in Jamaican organizations', *The Electronic Journal on Information Systems in Developing Countries*, Vol. 30, No. 4, pp.1–18.

Dubé, L. and Robey, D. (1999) 'Software stories: three cultural perspectives on the organizational practices of software development', *Accounting, Management and Information Technologies*, Vol. 9, No. 4, pp.223–259.

Dyba, T. and Dingsoyr, T. (2008) 'Empirical studies of agile software development: a systematic review', *Information and Software Technology*', Vol. 50, Nos. 9–10, pp.833–859.

Fink, A. (2009) *How to Conduct Surveys: A Step-by-Step Guide*, 4th ed., pp.1–136, Sage Publications, Inc., ISBN-10: 141296668X.

Fitzpatrick, R. (1996) *Software Quality: Definitions and Strategic Issues*, MSc dissertation, Staffordshire University, pp.1–35.

Hedelin, L. and Allwood, C.M. (2002) 'IT and strategic decision-making', *Journal of Industrial Management and Data System*, Vol. 102, No. 3, pp.125–139.

Ian, C. (1992) 'Using SEI maturity model to improve the software process', *Proceedings of the 6th Israel Conference on Computer Systems and Software Engineering*, IEEE Computer Society Prentice Hall.

Jarvinen, P. (2000) 'Research question guiding selection of an appropriate research method', *Proceeding of the 8th European Conference on Information Systems (ECIS 2000)*, Vienna.

Johnson, D.L. and Brodman, J.G. (1997) 'Tailoring the CMM for small businesses, small organizations, and small projects', *Software Process Newsletter, IEEE Computer Society Technical Council on Software Engineering*, Winter 1997, No. 8, p.1–6.

Jones, P. (2005) 'Process improvement in a small company', *International Journal*.

Jung, H. and Goldenson, D. (2009) 'Evaluating the relationship between process improvement and schedule deviation in software maintenance', *Information & Software Technology,* No. 2, pp.351–361.

Kan, S.H., Basili, V.R. and Shapiro, L.N. (1994) 'Software quality: an overview from the perspectives of total quality management', *IBM Systems Journal,* Vol. 33, No. 1, pp.4–9.

Kasekende, L. and Opondo, H. (2003) *Financing Small and Medium-scale Enterprises (SMEs); Uganda's Experience*, Bank of Uganda Working Paper, Research Department.

Kituyi, G.M. and Amulen, C. (2012) 'A software capability maturity adoption model for small and medium enterprises', *The Electronic Journal of Information Systems in Developing Countries*, Vol. 55, No. 1, pp.1–19.

Kituyi, G.M. and Tusubira, I. (2013) 'A framework for the integration of e-learning in higher education institutions in developing countries', *International Journal of Education and Development using Information and Communication Technology*, Vol. 9, No. 2, pp.19–36.

Kituyi, G.M., Rwashana, A.S., Mbarika, V. and Isabalija, S. (2012) 'A framework for designing sustainable telemedicine information systems in developing countries', *Journal of Systems and Information Technology*, Vol. 14, No. 3, pp.200–219.

Klein, H.K. and Myers, M.D. (2001) 'A classification scheme for interpretive research in information systems', in Trath, E.M. (Ed.): *Qualitative Research in IS*, Group Publishing, Hershey, PA.

Madachy, R.J. (2007) *Software process dynamics*, Vol. 24, No. 5, pp.511–523, IEE Press.

Mellon, C. (2007) 'Capability maturity model integration', *Lecture notes of Software Engineering*, pp.1–46, Software Engineering Institute.

Muller, S.D., Terma, A.S., Lystrup, D., Kraemmergaard, P. and Mathiassen, L. (2009) 'Managing cultural variation in software process improvement', *Engineering Management*, Vol. 56, No. 4, pp.584–599, IEEE.

Pankaj, J. (2001) 'The success of the SPI efforts in India, software quality professional', *IT Journal of American Society for Quality*, Vol. 2, No. 3, pp.251–258.

Pankaj, J. (2002) *Software Project Management in Practice*, Addison Wesley, Low Priced Edition by Pearson India.

Paulk, M.C. (2001) 'Extreme programming from a SW-CMM perspective', *XP for Universe,* Raleigh, NC.

Paulk, M.C., Weber, C.V., Curtis, B. and Chrissis, M.B. (1995) *The Capability Maturity Model: Guidelines for Improving the Software Process*, Addison-Wesley, Boston.

Persse, J.R. and Wiley, J. (2001) 'Implementing the capability maturity model', *The TQM Magazine*, Vol. 14, No. 2, pp.133–135.

Ravi, P. and Ruppel, C.P. (2010) 'A socio-technical approach to improving the systems development process', *Information Systems Frontier*, Vol. 12, No. 2, pp.19–234.

Richardson, I. (2001) 'SPI models: what characteristics are required for small software development countries?', *Software Quality Journal*, Vol. 10, No. 2, pp.100–113.

Trendowicz, M.K., Heidrich, J. and Lampasona, C. (2009) 'Measurement-based software quality evaluation', *Proceedings of the DASMA Software*, pp.3–16.

Appendix

Research questionnaire

Questionnaire on the critical parameters for adoption of SW-CMM in SMEs.

Introduction

This is an academic research on the factors that influence adoption of SW-CMM in SMEs.

You have purposely been identified as an important respondent, I request you to kindly spare some time to fill the questionnaire. The information you provide will be treated with utmost confidentiality and will be entirely for academic purpose.

Section A

Please place a tick against any option of your choice corresponding to the questions.

Qn.	*Question*	*Coding category*
Al	How long have you been in intensive software development?	
A2	How many employees are actively involved in software development.	
A3	Which of the following roles do you have in your in company? Tick the ones you have.	1 Senior manager
		2 Project manager
		3 Software project manager
		4 First-line software manager
		5 Software task leader
		6 Software engineering staff
		7 Software engineering group
		8 Software process engineering group
		9 Software configuration control board
		10 System test group
		11 Systems engineering group
		12 Software quality assurance group
		13 Contract management group
		14 Software subcontract management group
		15 Document support group
		16 Hardware engineering group
		17 Sales and marketing group
		18 Requirements management group
		19 Software estimating group
		20 Technology change management
		21 Training group
		22 Software configuration management

Section B

Please select an option which best represents your opinion by placing a tick against the number that corresponds to the answer of your choice as stated below:

5 = strongly agree; 4 = agree; 3 = uncertain; 2 = disagree; 1 = strongly disagree

Qn. no.	*Question*	*5*	*4*	*3*	*2*	*1*
B1	Some of the projects have been delayed and even cancelled					
B2	Customers often call requesting changes or modifications on software products.					
B3	User requirements are usually followed when developing software.					
B4	Corrective actions are taken when actual results deviate significantly from the software requirements.					
B5	Management accommodates software project failures					
B6	We follow some kind of process when developing software					
B7	Our staff are adequately trained to do their jobs successfully					
B8	The company ensures there is quality maintained during the process development.					
B9	Our clients are happy with our products					
B10	Our organisation has received some certification					

Section C

Please select an option which best represents your opinion by placing a tick against the number that corresponds to the answer of your choice as stated below:

5 = very important; 4 = important; 3 = fairly important; 2 = not important;
1 = not important at all

Qn. no.	*Question. Rate the following activities/practices according to their importance to your software development process.*	*5*	*4*	*3*	*2*	*1*
C1	Software requirements					
C2	Software project planning					
C3	Software project tracking and oversight					
C4	Software subcontract management					
C5	Software quality assurance					
C6	Software configuration management					
C7	Organisation process focus					
C8	Organisation product definition					
C9	Training programs					
C10	Integrated software management					
C11	Software product engineering					
C12	Inter group coordination					
C13	Peer review					
C14	Quantitative process management					
C15	Software quality management					
C16	Defect prevention					
C17	Technology change management					
C18	Process change management					

Section D

Please select an option which best represents your opinion by placing a tick against the number that corresponds to the answer of your choice as stated below;

5 = very influential; 4 = influential; 3 = fairly influential; 2 = not influential;
1 = not influential at all

Qn. no.	*Question. Which of the following factors could highly influence adoption of software process improvement approach?*	*5*	*4*	*3*	*2*	*1*
D1	Provision of adequate resources like software tools by management.					
D2	Management supports the budget for software improvement					
D3	Senior managements active participation in process initiatives					
D4	Managements emphasis on process improvement initiatives and benefits					
D5	Training all technical workgroups on process related activities and roles of (SPI)					
D6	Need to train software developers on software process improvement activities, new methods and tools.					
D7	Adoption of new technologies and tools by the organisation					
D8	Improving on employees competence in software development					
D9	Software developers initiating small changes					
D10	Appointing special individuals to coordinate the process					
D11	Identifying projects that can benefit from the change					
D12	Method of communication by ensuring real time feedback from people executing the process and having face to face talks with senior management					
D13	Having employees that spread that spread the process improvement vision, appointing special individuals to spear head the SPI process					
D14	Management constantly informing employees of changes in the process					
D15	Written organisational procedures followed when managing software development process					
D16	Organisation having clear and visible policies on SPI related activities, regular reviews of project performance and software plans					
D17	Rewarding the best performers by incentives like salary increments, bonuses					
D18	Organisational ability to define core strategy based on achieving operational excellence.					
D19	Management involving employees in decision making					
D20	Management's ability to choose appropriate improvement strategies					
D21	Active participation and involvement of all developers in the SPI initiatives					

End of Questionnaire
Thank you so much for Your time