A SOFTWARE CAPABILITY MATURITY ADOPTION MODEL FOR SMALL AND MEDIUM ENTERPRISES IN DEVELOPING COUNTRIES

Geoffrey M. Kituyi
Department of Business Computing
Makerere University Business School
Uganda
kimayoka@gmail.com

Christine Amulen
Department of Business Computing
Makerere University Business School
Uganda
camulen@mubs.ac.ug

Abstract
Controlling and improving the processes used to develop software has been proposed as a primary remedy to the problems of schedule, over budget and non conformity to requirements in software development. Consequently the software development community has adopted several Software Process Improvement approaches such as the Software Capability Maturity Model published by the SEI as a criterion to evaluate and improve software process maturity. The premise of the Software Capability Maturity Model is that a mature development process will deliver products on time, within budget, requirements and of high quality. Research carried out in developed countries to evaluate the effectiveness of such programmes has confirmed that Software Process Improvement can contribute to the production of high quality software. However, this phenomenon has been less studied in smaller enterprises such as those in Uganda and other developing countries. This study examined the effectiveness of the Software Capability Maturity Model for use in small software developing enterprises and established that the model was difficult to implement in such organizations. Many of its constructs were irrelevant. An adoption model for software capability maturity was designed based on only those factors and constructs of the Software Capability Maturity Model that were found to be relevant to Small and Medium Enterprises. The new model was tested using case study and expert judgment methods. The experts and software developers indicated that the proposed model is more applicable in Small and Medium Enterprises operating in resource constrained countries such as Uganda.

Key words: Capability Maturity Model, Software Development, Small and Medium Enterprises, Software Process Improvement, SMEs, Uganda, Developing Countries

1. INTRODUCTION
Software Process Improvement (SPI) is a systematic approach to improve software processes in organizations. This approach was developed by the Software Engineering Institute (SEI) inspired by the work of Humphrey (1989). The basic idea of SPI is to focus on software processes as social institutions with a complex interplay of people, methods, tools, and products (Aaen et al., 2000).

There are standard models such as the popular Software Capability Maturity Model (SW-CMM) that are acclaimed as the leading quality improvement standard that describes managerial processes that can be followed to attack software development difficulties (Visoiu, 2007).

Software Capability Maturity Model (SW-CMM) is defined by Dooley et al. (2001) as a methodology used to develop and refine an organization’s software development process. The model describes a five-level evolutionary path of increasingly organized and systematic processes. SW-CMM can also mean a process model that is used to measure
organizations Software Process Maturity and can be used for Software Process Improvement (Johnson & Brodman, 1999). The SW-CMM is portrayed as a descriptive, normative model that defines the key practices that characterize each successive level of process maturity. The model provides a method for assessing organizations’ capability to produce high quality software. According to Paulk (1993), the model consists of a group of key practices that are neither new nor unique to Software Engineering. These key process areas must be satisfied in order to achieve a given level of software maturity. In this paper, the SW-CMM and CMM are used to mean the same model.

The SW-CMM has been widely advocated for and adopted by many software developers as a means of improving software processes as well as a guide to quality software products (Paulk et al., 1995). However, Hadden (2001) observes that many software professionals believe the key practices listed in the SW-CMM are inappropriate and not scalable for Small Software Enterprises. Practitioners argue that the cost of implementing the model outweighs its benefits. According to Anderson et al. (2002), using SW-CMM to improve software processes is too cumbersome, time consuming and costly, especially for small enterprises. Some of the key practices in the SW-CMM appear inappropriate and un-scalable for Small Software Enterprises, yet Small Software Enterprises have as many concerns as those that affect the large ones. Detailed studies of the SW-CMM by Paulk (1993) reveal that it is very expensive for Small and Medium Enterprises (SMEs) to adopt the SW-CMM since its guidelines tend to be oriented towards large Software Enterprises. Most SMEs find the SW-CMM problematic to adopt, although its fundamental concepts are useful.

Therefore, significant factors of the SW-CMM in SMEs operating in developing countries needed to be established and a more suitable model for Small Software Enterprises designed. This would increase the likelihood of its adoption in Small and Medium Software Enterprises in developing countries like Uganda.

1.1 Statement of the Problems

According to Delroy and Duggan (2007), the software development global market is opening its doors to more developing countries, giving Small Software Enterprises an opportunity to benefit from increasingly competitive conditions. However, for Small Software Enterprises to benefit from these opportunities and become serious competitors, they must adopt and demonstrate internationally accepted Software Process Improvement (SPI) practices. Much as SC-CCM has been praised for SPI, Hadden (2001) notes that the guidelines prescribed in the SW-CMM are mainly aimed at supporting “large contracting Software Enterprises”. In many cases, these guidelines are not appropriate for direct use by SMEs involved in software development in developing countries. This is because of the differences between these two entities in terms of resources, knowledge and skills base, motivation, customer characteristics and infrastructure among others. Consequently the quality of software developed by SMEs in developing countries has remained unsatisfactory. Amidst this, there is increasing pressures from customers in developing countries for their software developers to produce high quality software systems (Delroy & Duggan, 2007).

Nevertheless, Curtis et al. (1992) and Paulk and Hyder (1997) believe that the SW-CMM can be tailored to fit the business application development segment of software markets and can be extended without difficulty to areas where a contract does not exist. It is also believed that there is a need to modify the SW-CMM to fit the small settings. This includes small and medium business organizations producing or acquiring software to support operations that will eventually be certified. Therefore, this study sought to develop a more applicable model that can be used by software developing SMEs in developing countries in SPI.
2. **LITERATURE REVIEW**

2.1 Concepts Related to Software Capability Maturity Model

Maturity endeavors started in the field of Software Engineering. The U.S Department of Defense (DoD) misjudged the ability of several of its contractors to develop mature software, so they deemed it imperative to mobilize Maturity Standards (Johnson & Brodman, 1999). The Software Engineering Institute (SEI) published the SW-CMM that can be used to measure organizations’ Software Process Maturity and can be used for Software Process Improvement as well as quality improvement. According to Paulk (2001), the Capability Maturity Model (CMM) is described as common-sense application of process management and quality improvement concept to software development and maintenance. The CMM is based on knowledge acquired from Software Process Assessments and extensive feedback from both industry and government. The CMM is believed to be the foundation for systematically building a set of tools which are useful in Software Process Improvement.

2.2 Maturity Levels of SW-CMM

Dooley et al. (2001) define maturity as “the extent to which a specific process is explicitly defined, managed, treasured, controlled and effective”. The SW-CMM describes an evolutionary improvement path for software development from an ad-hoc, immature process to a mature disciplined process in a path laid out in five levels known as Maturity Levels spelt out as follows: (i) **The Initial level** where only a few processes are defined and success depends more on individual effort than teamwork (Arent et al., 2000; Visoiu, 2007), (ii) **The Repeatable Level** where basic project management processes are implemented to follow cost, schedule and functionality (Abbassi, 2005) and organizations are more disciplined (Anderson et al., 2002), (iii) **The Defined Level**, where the standard process for developing and maintaining software across the organization are documented (Abbassi, 2005; Humphrey, 2002; Humphrey et al., 1999), (iv) **The Managed Level**, where detailed measures are taken on the process and products, and these are quantitatively understood and controlled (Ingalsbe et al., 2001; Hyder et al., 2006) and (v) **The Optimizing Level**, where the entire organization is focused on continuous process improvement, the organization has the means to identify weaknesses and strengthen the process actively with the goal of preventing the occurrence of defects (Horvat et al., 2000). Data on the effectiveness of the software process is used to perform cost benefit analysis of new technologies and proposed changes to the organization’s software process. At this level, continuous process improvement takes place using quantitative feedback. Herbsleb et al. (1997) established that feedback comes from the process and from piloting innovative ideas and technologies.

2.3 Benefits of Adopting SW-CMM to Organizations

Experience has shown that successful quality management must be oriented towards total process improvement rather than control of the final product alone. Much has been written about the short-term and long-term benefits of increasing Maturity Levels. Since the Capability Maturity Model was introduced in 1987, it has become a de-facto standard for assessing and improving software processes. The overall goal of the SPI is to help software organizations identify the key practices so as to increase the maturity of their software development process (Paulk & Hyder, 2007). Bollinger and McGowan (1991) argue that the concept of process or capability maturity is increasingly being applied to many aspects of product development, both as a means of assessment and as part of a framework for improvement. The SW-CMM can be used by an organization to plan improvements to its software process. It also reflects the needs of individuals performing Software Process Improvements, Software Process Assessments or Software Capability Evaluations.

Goldenson and Gibson (2003) report that the SW-CMM in some organizations is seen as a way of life that yields fantastic results. In many, it is a holy grail that will lead to a much
better way of life, yet to others, it is likened to “pouring sand into a gear box”, hence practical implementation is worth investigating.

Mondragon (2006) observes that some companies agree that savings significantly offset the costs of achieving and maintaining a maturity level. According to the published studies, the SW-CMM on balance can be considered a very successful model, particularly when combined with Total Quality Management (TQM) principles. Herbsleb et al. (1996) concur that the SW-CMM can be regarded important in contractual situations when suppliers must demonstrate their capability to design, develop, produce, install and service software.

### 2.4 Challenges SMEs Encounter in Using the SW-CMM

According to Delroy and Duggan (2007) evidence from SPI literature indicates that Small and Medium Enterprises that have attempted to adopt and apply SW-CMM have experienced several problems. They encounter difficulties as they struggle to adopt this model for use in their small projects (Anderson et al. 2002).

One of the first challenges for small organizations in using the SW-CMM for software improvements is that their primary business objective is to survive. Process improvement will help in finding the resources and assigning responsibility for process improvement. Following through by defining and deploying processes is a difficult business decision (Aaen et al., 2000).

Abrahamson (2000) established that employees of small organizations tend to believe that they are all competent people hired to do the job and they can not afford training in terms of either time or money. They also believe they all communicate well with one another because they are so close to one another; they are all heroes; they do whatever needs to be done; the rules do not apply to them; they just get in the way of getting the job done; they live with short cycle times and high stress. These attitudes and attributes have made it hard for them to adopt SW-CMM for SPI.

Garcia (2006) summarizes a broad range of challenges for process improvement in Small Software Enterprises, though many can be categorized as barriers to entry. These are; lack of core competences; failure to use basic technologies such as CMM and CMMI; inability to meet high costs of process improvement; existence of flat organizational hierarchies where roles and responsibilities are not clearly defined; lack of economies of scale to fund such projects through low cost capital and develop internal resources and core competence. Overcoming employees who are self-selected into small businesses because they want or expect to do their own things is also a big challenge.

Further to the above, Mondragon (2006) observes that small organizations find many of the SW-CMM practices irrelevant and even hard to implement. Many cannot afford the steep initial financial investment, high implementation costs, human resources and time commitment. In addition, smaller organizations have a lower capacity to absorb the unfavorable effects of attempted SPI implementation and have greater difficulty of recovering from failed SPI projects than their larger counterparts. The implementation problems that small organizations and SMEs encounter in their attempts to adopt the SW-CMM based SPI programs seem to resonate with the findings of (Anderson et al., 2002). Herrera and Ramirez (2003) and Mondragon (2006) all concur that the SW-CMM, though relatively prudent, may be too complex for small businesses and small projects. Orci & Laryd (2000) established that the publicly available models are not adequate for small organizations, one reason being that the number of roles proposed exceeds the number of employees in a small organization. Another reason is that the responsibilities and tasks proposed are too many and rigorous, at least, in very small organizations with only a few employees. Neither is that kind of rigor needed if it is easy to get oversight of the activities because of the easy communication and organizational structure.
It is believed that if small organizations from developing countries attempted to implement Software Process Improvement Models as they are currently structured, they would be subjected to all the difficulties highlighted above (Orci & Laryd, 2000). Yet small organizations just like large ones will have problems with undocumented requirements, the mistakes of inexperienced managers, resource allocation, training, documenting the product, limited effort equipment, material resources, highly centralized organizational structures with CEOs making most of the critical decisions coupled with strong cultural aversion to change (Herrera & Ramirez, 2003).

Despite these challenges, Orci & Laryd (2000) believe that, small organizations can be extraordinarily innovative and productive. Although there are massive problems that may require large numbers of people to solve, in general, small teams are more productive than large teams because they jell quicker and there are far fewer communication problems.

According to Johnson and Brodman (1999) all these factors have pushed developing countries away from the competitive business world. The SW-CMM is focused on large projects and large organizations; it seems that SW-CMM must become customized to small organizations both in scale and the application procedure. The experience of India shows that this standard causes higher confidence for foreign corporations (Huang & Han, 2006).

3. **Research Design**

This study can be classified as constructive research. Järvinen (2001) defines constructive research as typically involving the building of a new innovation based on existing research knowledge.

Though quantitative data were solicited for and used, the research design adopted in this study was mainly that of qualitative survey combined with conceptual analysis. Qualitative methods are believed to be suitable in this case because qualitative research is inclined to be deductive and is thought to be objective in testing theories.

According to Klein and Myers (2001), it is thought that in analyzing and interpreting qualitative data, the researcher can remain objective. Qualitative research adopts an interpretive stance of investigations, while attempting to understand phenomena through the meanings that people assign to them (Klein & Myers, 2001). In the study, the practical implications of SPI activities are considered vital components in understanding the meaning and their effect on the theoretical implications.

To help enrich on survey findings, interviews were conducted with selected respondents on the subject matter, literature review was carried out where current literature from journals, research articles, conference proceedings and other publications on Software Engineering where reviewed. These helped to accomplish the research objective of designing the new model.

3.1 **Sample Size and Sample Selection**

A sample of 150 respondents from 30 software developing SMEs in Uganda were selected to participate in the study, although responses from only 132 were analyzed after eliminating incomplete questionnaires. This was thought to give a reasonable degree of representation, given the fact that there were few software developers in Uganda. Some enterprises employed less than five people. The staff members were sampled using purposive sampling techniques.

In addition, a total of 30 respondents were purposively selected, 1 from each participating SME to participate in an interview. These were team leaders, supervisors and/or project managers. Out of the 30, only 13 interviews were successfully conducted.
Purposive sampling was chosen with the aim of including only those people of interest to the study. Purposive sampling was based on the fact that the sample was more knowledgeable on the issues of software development. This method is considered to be the most suitable in handling small populations according to (Klein & Myers, 2001).

The approach followed in developing the simplified Software Capability Maturity Adoption Model was basically the elimination and simplification of the key process area factors that influenced the SW-CMM based on the research findings from the survey.

4. DATA ANALYSIS AND PRESENTATION OF FINDINGS
Survey data were coded and analyzed using descriptive statistics including means, frequencies and percentages. These results were then presented using tables and figures in a summarized form. On the other hand, interview data were analyzed using content analysis method and presented qualitatively.

4.1 Validation of the Proposed Model
Validation was done on the model to establish its reliability and viability for adoption. One way of validating a framework or model is against the knowledge of experts in the field as Inglis and Alcock (2012) and Polanyi (1966) posit that experts have a wide, implicit and sometimes explicit understanding acquired over a long period of time.

The research sought the wide knowledge of ten experts from reputable IT firms and software developing organizations. According to Inglis and Alcock (2012), the larger the panel, the more reliable and credible will be the decision arrived at. Use of experts offers an opportunity of a wide rich knowledge on the subject matter. Credible results will be determined by the composition of the panel, penal briefing on the subject matter and what the validation process is intended for. The outcome of the validation was re-submitted to the experts for review. In this study, experts in the area of software development were given the model to test on their projects. Thereafter, they provided feedback based on their experiences. Their responses were grouped, coded and analyzed to make meaning out of them. Expert judgment validation exercise was conducted on the proposed model using the following parameters:

(i) Issues measured; the issues included the structure of the model, looking at its key process areas and the associated goals and objectives.

(ii) Instrument; instruments form the technique used to capture information from the evaluation exercise. The technique that was used to validate the model was use of self administered questionnaires designed with the help of experts.

(iii) Analysis; information obtained from the validation process was subjected to analysis in order to draw a true meaning and implications from data collected. This was later presented using statistical analysis charts and tables.

4.2 Case Study Validation Method
In addition to expert judgment, a case study was conducted on the improvement of the Education Information System (EIS) by Eminence Uganda Ltd at Makerere University Business School. The details and results from this exercise are presented later in this paper under validation results.

5. FINDINGS
Data on the current situation of software development in the organizations were collected on a five point agree-disagree scale. The results (see Table 1) show that respondents strongly agreed on the following; some of the projects have been delayed and even cancelled (mean=4.66); customers often call, requesting changes or modifications on software products

The Electronic Journal on Information Systems in Developing Countries
http://www.ejisdc.org
(mean=4.51); user requirements are usually followed when developing software (mean=4.01). However, the respondents strongly disagreed that management accommodates software project failures (mean=2.22), staff are adequately trained to do their jobs successfully (mean=2.74) and also that their organizations had received some quality certification (mean=2.10). Respondents were unsure on the following; corrective actions are taken when actual results deviate significantly from the software requirements (mean=3.21); we follow some kind of process when developing software (mean=3.85); the company ensures there is quality maintained during the process of software development (mean=3.88); our clients are happy with our products (mean=3.45).

Table 1: Analysis of the Current Situation in Software SMEs
Note: 5=strongly agree; 4=agree, 3=unsure, 2=disagree and 1=strongly disagree

<table>
<thead>
<tr>
<th></th>
<th>N</th>
<th>Mean</th>
<th>SDV</th>
</tr>
</thead>
<tbody>
<tr>
<td>Some of the projects have been delayed and even cancelled</td>
<td>132</td>
<td>4.66</td>
<td>.806</td>
</tr>
<tr>
<td>Customers often call requesting changes or modifications on software products.</td>
<td>132</td>
<td>4.51</td>
<td>.022</td>
</tr>
<tr>
<td>User requirements are usually followed when developing Software</td>
<td>132</td>
<td>4.01</td>
<td>.342</td>
</tr>
<tr>
<td>Corrective actions are taken when actual results deviate significantly from the software requirements</td>
<td>132</td>
<td>3.21</td>
<td>.065</td>
</tr>
<tr>
<td>Management accommodates software project failures</td>
<td>132</td>
<td>2.22</td>
<td>.412</td>
</tr>
<tr>
<td>We follow some kind of process when developing software</td>
<td>132</td>
<td>3.85</td>
<td>.098</td>
</tr>
<tr>
<td>Our staff are adequately trained to do their jobs successfully</td>
<td>132</td>
<td>2.74</td>
<td>.981</td>
</tr>
<tr>
<td>The company ensures there is quality maintained during the Process development</td>
<td>132</td>
<td>3.88</td>
<td>.012</td>
</tr>
<tr>
<td>Our clients are happy with our products</td>
<td>132</td>
<td>3.45</td>
<td>.146</td>
</tr>
<tr>
<td>Our organization has received some certification</td>
<td>132</td>
<td>2.10</td>
<td>.434</td>
</tr>
</tbody>
</table>

6. ANALYSIS OF THE FACTORS INFLUENCING SW-CMM ADOPTION

Descriptive means were used to determine the factors that influenced successful SW-CMM adoption in SMEs. Data were collected using a five point scale anchored from very influential to not at all influential. The results are presented in Table 2 and show that the most influential factors for SW-CMM adoption are; senior managements active participation in process initiatives (mean=4.69); managements emphasis on process improvement initiatives and benefits (mean=4.44); training all technical workgroups on process related activities and roles of SPI (mean=4.56); need to train software developers on Software Process Improvement activities, new methods and tools (mean=4.42); method of communication by ensuring real time feedback from people executing the process and having face to face talks with senior management (mean=4.44); management’s ability to choose appropriate improvement strategies (mean=4.37); management constantly informing employees of changes in the process (mean=4.01); organization having clear and visible policies on SPI related activities, regular reviews of project performance and software plans (mean=4.23); rewarding the best performers with incentives like salary increments, bonuses (mean=4.73); management involving employees in decision making (mean=4.21). These factors were grouped and summarized (see Table 3).
Table 2: Factors Influencing SW-CMM Adoption

Note: 5=very influential; 4=influential, 3=fairly influential, 2=not influential and 1=not influential at all.

<table>
<thead>
<tr>
<th>Factor</th>
<th>N</th>
<th>Mean</th>
<th>SDV</th>
</tr>
</thead>
<tbody>
<tr>
<td>Provision of adequate resources like software tools by management.</td>
<td>132</td>
<td>2.07</td>
<td>.321</td>
</tr>
<tr>
<td>Management supports the budget for software improvement</td>
<td>132</td>
<td>2.89</td>
<td>.136</td>
</tr>
<tr>
<td>Senior managements active participation in process initiatives</td>
<td>132</td>
<td>4.69</td>
<td>.024</td>
</tr>
<tr>
<td>Managements emphasis on process improvement initiatives and benefits</td>
<td>132</td>
<td>4.44</td>
<td>.073</td>
</tr>
<tr>
<td>Training all technical workgroups on process related activities and roles of (SPI)</td>
<td>132</td>
<td>4.56</td>
<td>.045</td>
</tr>
<tr>
<td>Need to train software developers on Software Process Improvement activities, new methods and tools.</td>
<td>132</td>
<td>4.42</td>
<td>.111</td>
</tr>
<tr>
<td>Adoption of new technologies and tools by the organization</td>
<td>132</td>
<td>4.13</td>
<td>.024</td>
</tr>
<tr>
<td>Improving on employees competence in software development</td>
<td>132</td>
<td>3.64</td>
<td>.327</td>
</tr>
<tr>
<td>Software developers initiating small changes</td>
<td>132</td>
<td>2.33</td>
<td>.139</td>
</tr>
<tr>
<td>Appointing special individuals to coordinate the process</td>
<td>132</td>
<td>2.88</td>
<td>.026</td>
</tr>
<tr>
<td>Identifying projects that can benefit from the change</td>
<td>132</td>
<td>3.25</td>
<td>.067</td>
</tr>
<tr>
<td>Method of communication by ensuring real time feedback from people executing the process and having face to face talks with senior management</td>
<td>132</td>
<td>4.44</td>
<td>.032</td>
</tr>
<tr>
<td>Having employees that spread the process improvement vision, appointing special individuals to spear head the SPI process</td>
<td>132</td>
<td>2.88</td>
<td>.037</td>
</tr>
<tr>
<td>Management constantly informing employees of changes in the process</td>
<td>132</td>
<td>4.01</td>
<td>.653</td>
</tr>
<tr>
<td>Written organizational procedures followed when managing software development process</td>
<td>132</td>
<td>2.57</td>
<td>.564</td>
</tr>
<tr>
<td>Organization having clear and visible policies on SPI related activities, regular reviews of project performance and software plans</td>
<td>132</td>
<td>4.23</td>
<td>.083</td>
</tr>
<tr>
<td>Rewarding the best performers by incentives like salary increments, bonuses</td>
<td>132</td>
<td>4.73</td>
<td>.231</td>
</tr>
<tr>
<td>Organizational ability to define core strategy based on achieving operational excellence.</td>
<td>132</td>
<td>2.67</td>
<td>.087</td>
</tr>
<tr>
<td>Management involving employees in decision making</td>
<td>132</td>
<td>4.21</td>
<td>.124</td>
</tr>
<tr>
<td>Management's ability to choose appropriate improvement strategies</td>
<td>132</td>
<td>4.37</td>
<td>.056</td>
</tr>
<tr>
<td>Active Participation and involvement of all developers in the SPI initiatives</td>
<td>132</td>
<td>3.76</td>
<td>.071</td>
</tr>
</tbody>
</table>
Table 3: Key Factors for SW-CMM Adoption

<table>
<thead>
<tr>
<th>No</th>
<th>Factor</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Management’s commitment</td>
</tr>
<tr>
<td>2</td>
<td>Organization’s culture</td>
</tr>
<tr>
<td>3</td>
<td>Effective communication</td>
</tr>
<tr>
<td>4</td>
<td>Having reward systems</td>
</tr>
<tr>
<td>5</td>
<td>Training</td>
</tr>
<tr>
<td>6</td>
<td>Developer’s involvement</td>
</tr>
<tr>
<td>7</td>
<td>Having people to champion SPI</td>
</tr>
<tr>
<td>8</td>
<td>Management’s ability to choose an appropriate improvement strategy</td>
</tr>
</tbody>
</table>

6.1 Explanation of the Factors for SW-CMM Adoption

In the analysis of the data collected, the major significant factors were identified as follows:

6.1.1 Management’s Commitment

Senior managers must absolutely be convinced about the initiatives’ relevance from the beginning. Management commitment and support helps throughout the project by providing all the necessary resources required for implementing Software Process Improvement. Often initiatives not supported by management may not be successful. From the interviews, most respondents (9 out of 13) indicated that software projects that failed in their organizations did not have adequate top management support. The following statement was given by a supervisor:

“...most of the projects in my organization that failed did not have adequate support of management, both on the side of clients and sometimes our own managers. This is because top management makes key financing decisions for such projects.”

6.1.2 Management’s Ability to Choose an Appropriate Improvement Strategy

The above factor was considered the most critical factor which was arrived at by combining two factors; management’s emphasis on process improvement and the ability to choose appropriate improvement strategies. Management’s ability to determine the appropriate SPI strategy is believed to be the key step in the process of founding a mature organization. Management must be able to choose an appropriate model that can be applicable to a particular size of the organization. While most people interviewed were skeptical about management’s ability to choose an appropriate improvement strategy, citing reasons such as dynamisms in software developments environments and the requirements, some indicated that once management has chosen an appropriate strategy, this influenced success. A software project manager stated as follows:

“The role of management in choosing software improvement strategies cannot be underestimated. ...some of us entirely rely on what management tells us to do; therefore management must be capable of choosing the right improvement strategy”
6.1.3 Developer’s Involvement
Involving all software development personnel in SPI initiatives is very important. This can be done by encouraging all of them to choose a key process area that they are comfortable doing and forming workgroups to address the key parameters for software improvement. The interviewees perceived developers’ involvement as not really being important. Instead 8 out of the 13 suggested that the client should be involved since they are the consumers.

“...we the developers are involved in software improvements by default. However, this is not the case with clients. Sometimes, we tend to assume that we know what the customer wants and end up with systems that are more developer centered”

6.1.4 Effective Communication
This represents all methods of communication. Software developing organizations should establish effective channels of communication between the workgroups and customers. Receiving real-time feedback from people who actually execute the processes being improved helps to deploy effective process improvements. The findings from interviews entirely agreed with effective communication as a key for SW-CCM adoption. In fact, all the 13 respondents interviewed indicated that effective communication contributed 90% toward success:

“Communication has always been a key determinant for the success of software projects I have participated on. There is need for us to communicate well with clients, first of all in raising and refining the requirements and secondly in testing the developed software. Sometimes the clients don’t like our products but as long as they have not communicated, there is no way we can make improvements.”

6.1.5 Organizations’ Culture
This is the second-most critical factor which emphasizes the need for organizations to have visible policies on SPI, regular reviews of project performance and software plans. When SMEs are implementing and deploying Software Process Improvement strategies, it’s very important to put into consideration the organizational culture. Different organizations have different beliefs and cultures. For instance, Small Software Enterprises strongly believe that, the SW-CMM is not meant for them, while to large software houses, such connotation may have an impact in the event of trying to adopt SW-CCM. In agreement with this factor, a software project Team Leader stated as follows:

“...this is where most of the differences between large and SME software developing organizations are. While large organizations tend to have a more formalized and well known organizational culture, most SMEs do not. This is even worse in developing countries, where in most cases; organizational culture is influenced by an individual who wields power. I think SMEs should work towards harmonizing this issue.”

6.1.6 Having Reward Systems
Rewarding best performers in the organization through incentives such as salary increments, bonus payments can increase the motivation of such individuals. Taking an individualist focus like, giving incentives to individuals for performing process improvement related activities can play a big role in encouraging staff to take plays significant roles in SPI. An
organization must define recognition and reward systems based on tangible value of the employees to an organization.

Much as this was highlighted as an important factor many interviewees indicated that there were no reward systems in place. They called for SMEs to establish a reward system for outstanding employees. The following was extracted from a Project Manager:

“To me, success does not really matter because, other than safeguarding my job there is no other benefit. This is even worse with team members who do not have management positions. They feel cheated every time we successfully accomplish a task and they gain nothing out of it other than their monthly salaries.”

In the same development, a supervisor expressed her dissatisfaction with their current reward system as follows:

“When our company successfully handles a big project, our bosses pay themselves bonuses, ignoring that it is us who did the work. We also need to have a fair share of success; just the same way we are blamed when projects fail.”

6.1.7 Training
The need to train software developers on SPI activity, new methods and tools was highlighted as being very important. There is need to undertake process-related training. All developers in the software development unit need to be trained on all the processes and roles of the SW-CMM. While interviewees agreed that continuous training is important for successful software development and improvement, most (11 out of 13) decried the lack of training opportunities in their organizations. The following statement was captured from one of them:

“Nowadays, they call it quality at the gate”, literally meaning employees are recruited when they have all the necessary skills and qualifications and therefore there is no need for employers to spend money training them.

“However, this is not the case especially, considering the ever-changing software development environment...our employers should know that when we are well trained, it helps us improve on the quality of service. This translates into better products, increased customer satisfaction and hence profitability. It is not only for us.”

6.1.8 Having People to Champion SPI
Having employees that take the initiative of spreading the SPI vision, goals and activities personally improves adoption. It’s believed that an effective way to convince everyone of the importance of improving the software process was to have SPI champions on the software development team. The findings from interviews also agreed with this as 8 out of 13 interviewees urged for project champions i.e. the people who advocate for process improvement.

Figure 1 shows the key parameters for adoption of the SW-CMM. It shows that out of the overall eight factors identified in the analysis, two factors were seen as significant with highest scores, i.e. management’s ability to choose appropriate improvement strategies and organization cultures. Therefore, they can be considered more critical for adoption of the CMM in the case of small software developing enterprises. These factors are incorporated in the new model for adoption of SW-CCM seen in Figure 2.
Figure 1: Key Parameters for Adoption of the SW-CMM
6.2 Application of the Model

The improved Software Capability Maturity Adoption Model proposed for use in small software developing enterprises includes only three levels of SW-CMM as opposed to five levels specified in the original SW-CMM. Level one is where most Ugandan Software Enterprises are estimated to be. The long term objective here is to target progression to level 2 and 3 and if necessary gradually introduce more advanced practices associated with higher levels as the enterprises grow. The original model has a total of 18 factors for the level. However, these have been reduced to 8. Level 2 (Repeatable level) of the original model has 6 factors, which have been reduced to 4, while Level 3 (Defined level) in the original model had 7 factors that are now 2. This gives a total of 8 key factors for adoption of SW-CCM which are more manageable by SMEs in developing countries like Uganda.

6.3 Validation of the Model

A case study was conducted at Makerere University Business School, where a software developing SME called Eminence Uganda Ltd had been contracted to develop and deploy an Education Information System (EIS) that would process and manage staff and students data. Eminence has a team of 12 software developers specializing in different fields namely systems analysis and design, coding, software project management, requirements analysis and business process re-engineering among others.

Initially there was no need for students to register online using the EIS. Students would visit the Registrar’s Office with their documents and get registered manually. They would then be issued with registration cards. In addition to this, the system did not have a
provision for students to view and print their tuition statements. The system did not also capture and process students’ results.

However, 2 years after implementing the systems, new processes were introduced in the business school. The students were required to make payments, print tuition statements and register online using the EIS. Lecturers were also required to enter students’ results including all coursework, continuous assessments and final exam marks online via the EIS. The students would then access and print their results online. These new developments meant that the existing EIS had to be re-designed and improved to cater for the new processes. Consequently, Eminence Uganda Ltd deployed a team of 2 software developers, 2 systems analysts and 1 business process re-engineering expert to study the new business processes and come up with an appropriate solution that would culminate into improvement of the EIS.

6.4 Use of the Model for Adoption of the SW-CMM for Improving the EIS

Faced with the above daunting tasks, the researchers presented the new Model for adoption of the SW-CMM to the team. The team were taken through the various steps of the model and shown how to use them in improving software processes. They were divided into 3 smaller groups and given different tasks on new processes to accomplish while using the new model. Each group came up with new software designs, based on their tasks. These were then harmonized at team level and implemented. Having used the model, the team were then asked to give their experiences about it and show whether and how the model helped them execute their tasks. These responses are presented as follows.

6.4.1 Clarity and Simplicity of the Steps in the Model

Most of the team members (4 out of the 5 members) indicated that the model was simple, clear and easy to understand and use. The members believed the model had been designed with diligence and due consideration for different categories of software development teams. The system designers particularly applauded the model for having satisfied the basic principles of design. The following statement was recorded from one of them:

“... the eight factors influencing adoption have been well-arranged in their order of significance. At the beginning, all factors are prevalent. However, these keep reducing in each and every successive phase toward adoption”

Although programmers usually pay little attention to design issues, a programmer stated as follows:

“This model is easy to use compared to other complex ones. One does not need to read a textbook for them to be able to use it”

6.4.2 Time Management

The model helped the software team reduce the time spent on the job from the 7 days that had been anticipated to 5 days. The members revealed that they usually spend not less than a week on such tasks; however, having followed a scaled-down model, they were able to save time.

6.4.3 Cost Saving

Having been able to reduce on the time spent on work, the model helped the team save costs. The model also meant that even individuals with little experience and expertise in Software Process Improvement can perform the tasks since the model gives them clear guidelines. The important stakeholders to be enlisted for consultation are presented in the model. The team therefore expressed hope that in future projects, they would send fewer people to do the work. This would help reduce on costs.
6.4.4 Software Quality Improvement

The team envisaged development of better quality software systems in future once they used the new model.

“Once top management comes up with a good process improvement strategy and is fully committed to implementing it, there are higher chances of producing better products. The model also talks about total involvement of stakeholders in the software development process. All these translate into better quality”

6.5 Validation Results from Experts

6.5.1 Description of Expert Characteristics

In addition to case study validation above, experts were given the model to interact with and give their opinions. The panel of experts comprised of Software Developers (40%), System Designers (30%), System Engineers (20%), IT Managers and Technical Support Personnel (10%). 40% of the experts had working experience of 4–6 years, 30% had worked for 7-9 years, 20% had worked for over ten years and 10% had worked for 1-3 years. In addition, 50% of the experts involved in the evaluation study had done 6-10 software projects, 30% had done 11-15 projects while 10% had done 1-5 projects and more than 15 projects. 90% of the experts had knowledge on SPI, while 10% had no clue of SPI. It was also established that most of the experts i.e. 90% had knowledge of the SW-CMM while the 10% knew only the ISO9000 SPI model.

From this analysis it can be deduced that the panel constituted of knowledgeable, well experienced experts in the field of software development. Therefore, they exercised an acceptable degree of authority on the subject matter, hence suitable to validate the model.

6.5.2 Summary of the Validation Results from Experts

Experts were asked to validate the model in various key areas. There responses were coded and analyzed as seen in Figure 3:
6.6 Levels Included in the Modified Capability Maturity Model

The validation process was based on the structure of the model that was composed of Levels, Key Process Areas (KPAs) and the logical flow of activities, objectives and the adoptability of the model. As seen in figure 3, 70% of the experts strongly agreed that the model included a reasonable number of levels as compared to the original model; while 65% concurred that the levels in the model were simple and easy to understand.

6.6.1 Retained Key Process Areas

Figure 3 shows that 65% of the experts strongly agreed that KPAs retained in the model were adequate for SMEs considering their size. 50% strongly accepted that the KPAs captured the key issues in the software development process. 55% strongly accepted that the KPAs also presented a logical flow of information. 71% concurred that the objectives associated with the KPAs were clear and correct and can be achievable. Considering the KPAs retained in the model, the experts agreed that they were satisfactory and viable and can be easily applied by SMEs in developing countries.

6.6.2 Adoptability of the Model

In addition, Figure 3 shows that 77% of the experts agreed that the model was adoptable due to its logical flow of activities and the general structure of the model reflects a complete Software Process Improvement and also that it can be considered viable and suitable for adoption, especially for SMEs in developing countries. 70% concurred that since the model was comprehensive in nature, it was viable for adoption. 75% strongly agreed that the model if followed systematically was likely to result into high quality software with minimum cost.
and timeliness. This is so because the model reflected complete software development process improvement. Therefore, it can be recommended for adoption by SMEs in developing countries like Uganda.

7. **CONCLUSION**

According to the results of the validation process, the proposed capability model is credible, viable, realistic and less cumbersome for SMEs in developing countries. It can be implemented by SMEs with minimal resources. This model is developed after a critical study of various models, their strength and weaknesses. Having obtained favorable responses from the validation case study, the model was later subjected to scrutiny by a panel of experts who proved its credibility, viability and completeness. Majority of the experts believed the proposed model was applicable to SMEs in developing countries.

Muller et al. (2007) argues that controlling and improving the processes used to develop software has been proposed as a primary remedy to the problems of schedule, over budget and non conformity to requirements in software development. Consequently the software development community has to adopt at least one Software Process Improvement (SPI) approach to remain competitive in the software industry. This model presents a more suitable alternative for SMEs in developing countries.

7.1 **Recommendations**

The ability to produce reliable and usable software within time and budget is a requisite in this field. Furthermore, customer satisfaction has become a most important driving force for many software development organizations. However, software user requirements have become very dynamic and customers keep changing their needs.

In light of the above observations, this study recommends that, Small and Medium Software Enterprises adopt the proposed model for SPI initiatives that increase an organization’s Software Process Maturity and quality. Software developers should focus on a limited set of activities as specified in the modified model and working towards achieving them. This will enable Small Software Enterprises to steadily improve their software processes to enable continuous and lasting gains in the software development.

Since it is believed that it is realistically impossible to create a comprehensive system for Total Quality Management without a universally acceptable model to base it on, Small Software Enterprises should adopt SPI as a competitive strategy for their regions and must be supported by their governments. According to Glazer (2006), Small Software Enterprises are becoming aware of the opportunities and challenges of process improvement.

7.2 **Areas for Future Research**

Further research should focus on formulating a research model, which illustrates the relationship between the key software practices proposed in the simplified model and the quality of software products produced and the success of Software Process Improvement in small settings particularly in developing economies. There is also a need for software developers to test this model using real-life software projects in different settings. This will help generate comparative data for evaluating applicability of this model.

8. **REFERENCES**


