2008

Identifying Key Software Development Practices in the English-Speaking Caribbean Using the Nominal Group Technique

Delroy A. Chevers
Department of Management Studies The University of the West Indies, delroy.chevers@uwimona.edu.jm

Stanford E. Moore
Deans Office, Faculty of Social Science The University of the West Indies, stanford.moore@uwimona.edu.jm

Evan W. Duggan
Mona School of Business The University of the West Indies, evan.duggan@uwimona.edu.jm

Annette M. Mills
Department of Accountancy, Finance and Information Systems College of Business and Economics, University of Canterbury, annette.mills@canterbury.ac.nz

Identifying Key Software Development Practices in the English-Speaking Caribbean Using the Nominal Group Technique

Delroy A. Chevers
Department of Management Studies
The University of the West Indies
Email: delroy.chevers@uwimona.edu.jm

Stanford E. Moore
Deans Office, Faculty of Social Science
The University of the West Indies
Email: stanford.moore@uwimona.edu.jm

Evan W. Duggan
Mona School of Business
The University of the West Indies
Email: evan.duggan@uwimona.edu.jm

Annette M. Mills
Department of Accountancy, Finance and Information Systems
College of Business and Economics, University of Canterbury
Email: annette.mills@canterbury.ac.nz

Abstract

This paper explains how a simplified process improvement framework was developed with practices from the capability maturity model integration (CMMI) model using the nominal group technique (NGT). The NGT was used to generate consensus on key software process improvement practices that are likely to lead to the improvement of the quality of information systems (IS) in the English-speaking Caribbean (ESC). We present the approach taken, the results derived, and the benefits of using this approach. NGT sessions, involving 30 IS professionals were conducted in four ESC territories to reduce and finalize a list of key software development practices that would be more suitable for ESC nations to focus on.

Keywords

Capability maturity model integration, software development practices, English-speaking Caribbean, nominal group technique, small organization

INTRODUCTION

In a study conducted by Chevers & Duggan (2007) it was discovered that a large majority of Jamaican software development firms do not use any formal assessment methods to determine the maturity of their software production process or adopt any software process improvement (SPI) programs. SPI is a series of actions taken to identify, analyze and improve the existing software production processes within an organization. Maturity is defined as the degree to which a process is defined, managed, measured, and continually improved (Dooley et al. 2001). The 2007 study focused on software development firms who produced software products for their internal use as well as those who produced software products for sale.

The possible reasons for the lack of SPI focus in Jamaican firms are unclear. However, based on findings in developed countries, it is revealed that SPI models are not developed with the needs and constraints of small firms needs in mind (Herrera & Ramirez, 2003). As a result SPI implementation can be cumbersome, time consuming, disruptive and costly for firms in Latin America (Buchman & Bramble, 1995; Turner, 2007). Turner (2007) also asserted that for many small software development firms the amount of time, effort, cost and culture shock associated with implementing SPI practices, (in this case CMM/CMMI processes) can be overwhelming. Similarly, (Staples et al. 2007) found that small firms tended not to adopt such practices because of factors such as the high cost, limited time available and size.

We have therefore extended this notion of SPI models not being developed for small firms to include software development firms in the English-speaking Caribbean (ESC) based on their size. In this study, we adopted the European Commission definition of small enterprises in which the number of employees in the firm is ten to

forty-nine and the turnover is between EU$2 - 10 million. Based on this definition, most or all software development firms in Jamaica and the wider ESC would be classified as small. The English-speaking Caribbean refers to members of the Caribbean Community of Common Market (CARICOM) whose first language is English (Duggan & Virtue, 2004) of which there are seventeen countries.

Irrespective of whether a firm is large or small, customers are demanding high quality software (Chevers & Duggan, 2007), that is, software that produces the required features, are relatively easy to access and those that retains business relevance beyond deployment (Duggan, 2003). However, the information systems (IS) literature supports the notion that insufficient attention to SPI practices can affect firm's ability to supply quality software (Bicego & Kuvaja, 1996; Hasse, 1996; Krishnan & Keller, 1999). It is assumed that small software development firms in Jamaica and the ESC would like to improve their delivery process and by extension increase the possibility of producing higher quality software products. Given the difficulty experienced by small firms in implementing SPI processes such as the CMM/CMMI (Staples, et al. 2007; Turner, 2007), it seems reasonable to suggest that a simplified process improvement model with practices that are applicable to and consistent with the norms and culture of small firms, in particular those in our research context of the ESC, would increase the likelihood of adoption and use of SPI programs in these firms. In an effort to increase the likelihood of adoption and use of SPI initiatives in the ESC, it is proposed that this study adapts an existing process improvement framework – the capability maturity model integration (CMMI) to suit the norms and constraints of small software development firms in the ESC. Hence our research question is "What SPI practices do IS professionals consider applicable and relevant for firms in the English-speaking Caribbean based on their culture and constraints?" The outcome sought in this study is to identify the set of most highly ranked SPI practices considered relevant to small software development firms, which will be incorporated into our proposed process improvement framework. The nominal group technique (NGT) was used among thirty information systems professionals in four ESC countries to identify these highly ranked SPI practices.

The rest of the paper is organized as follows: A review of the relevant SPI literature is presented in the next section, followed by a review of the NGT literature. This is followed by the methodology, then the result of our findings and discussion, and finally a conclusion is offered.

SOFTWARE PROCESS IMPROVEMENT (SPI)

It is widely accepted by some scholars (Paulk et al. 1993) in the IS community that the quality of a product is largely determined by the quality of the process used to produce it (Deming, 1986). This premise borrowed from the manufacturing sector, has lead to the birth of software process improvement in which extensive focus is placed on establishing process stability, maturity and capability in an effort to reduce variability in the process. This by extension is hoped to increase the likelihood of producing high quality software products. This is achieved through a 2-step approach as distilled by Humphrey (1989) which is (1) assessing the maturity and discipline of a firm's existing software development practices through process improvement frameworks like the capability maturity model integration (CMMI) and (2) embarking on implementation plans to achieve higher process maturity.

Information systems researchers have corroborated the notion that software quality is positively correlated with the quality of the IS delivery process and that SPI effort is a necessary, though not sufficient, condition for the production of high quality systems (Ravichandran & Rai, 2000). In fact, the prior research (Dooley et al. 2001; Harter et al. 2000; Herbsleb et al. 1997; Johnson & Brodman, 1996; Krishnan & Keller, 1999) has demonstrated the benefits of software process assessment and SPI efforts in areas such as (1) improved software quality, (2) reduced project cycle time, (3) reduced development cost, (4) improved productivity, and (5) improved customer satisfaction.

There are numerous software process improvement models including the Software Capability Maturity Model (SW-CMM), SE-CMM produced by Enterprise Process Improvement Collaboration (EPIC), Software Process Improvement Capability determination (SPICE), Bootstrap, Personal Software Process (PSP), Team Software Process (TSP) and Capability Maturity Model Integration (CMMI) the successor of SW-CMM.

Of these models, the CMMI process improvement framework was selected for this study because:

- It is viewed as the most widely known method for SPI (Batista & de Figueiredo, 2000)
- It is designed to be tailored and adapted to meet specific organizational context and needs because it is a normative model (Paulk et al. 1995). There are about 34 CMMs developed by different groups (Reifer, 2000)
- It is the most used model for judging the maturity of software processes (El Emam & Madhavji, 1995)

- It is the leading quality improvement standard in North America for software development (Persse & Wiley, 2001).
- It is based on best practices derived from many years of empirical research, and has a proven record of achievement and it is goal focused (Beecham et al. 2005)
- It is the most widely used model in the world (Weber & Layman, 2002)
- Its approach to software process improvement is the most dominant paradigm of organizational change that software firms implement (Ngwenyama & Nielsen, 2003)
- It has become a de facto standard for assessing and improving processes (Rogoway, 1998)
- It is a 'living model' that is actively supported by the SEI who continually upgrade the model in response to the needs of the software industry (Beecham et al., 2005)

The CMMI defines the best practices of a company, by assessing the state of the company and prescribing improvement practices. It is more than just an assessment instrument. The framework describes an evolutionary improvement path through 5 levels or stages beginning with an ad hoc, immature process and progressing to a mature, disciplined and capable process.

- At level 1, the *initial stage*, the developmental process is described as chaotic. Project success depends on individual effort and heroism. There is no structured management of the software development process and knowledge transfer from one project to the next is haphazard.
- At level 2, the *repeatable stage*, there exist embedded project management practices which enhance the ability to repeat previous successes with similar projects.
- At level 3, the *defined stage*, the software process is standardized, documented and communicated to key personnel.
- Level 4, the *managed stage*, takes on a quantitative approach to measure the effectiveness of process improvement.
- Level 5, the *optimizing stage*, seeks to create a culture of continuous process improvement through quantitative feedback.

The CMMI Version 1.1 (2001) and Version 1.2 (2006) is a product suite with a full collection of models, training materials and appraisal methods (SEI, 2006). All key process areas and measurement practices within the SW-CMM are included in the CMMI (Weber & Layman, 2002), because the CMMI was built on and has extended the best practices and lessons learned from the SW-CMM and other process improvement models (SEI, 2004). In general, the CMMI process areas can be classified as (1) project management, (2) process management, (3) engineering and (4) support.

In order to promote the adoption and use of SPI interventions in the ESC, this study adapts a leading and popular process improvement framework - the CMMI to the study context. This adaptation will take the form of identifying and incorporating relevant and applicable SPI practices into a modified process improvement framework, which recognizes constraints of the ESC such as limited resources in finance, labor, equipment and material (Thong et al. 1996) and other cultural factors such as aversion to change and low productivity (Herrera & Ramirez, 2003).

The CMMI framework has five maturity levels consisting of twenty-two process areas or practices. However, we are taking a phased approach in this exercise by firstly exposing IS developers and practitioners to levels 1, 2 and 3 which has eighteen practices altogether, and then in the near future to incorporate the more advanced practices at levels 4 and 5. In determining appropriate modifications we sought the assistance of IS experts (analysts, developers and programmers) in four ESC countries namely, Guyana, Trinidad, Barbados and Jamaica to identify key software development practices in their respective countries. The goal of this stage of the research was to identify the most highly ranked practices in each of the four countries; the nominal group technique (NGT) was used to enable this process.

THE NOMINAL GROUP TECHNIQUE (NGT)

For many years freely interacting groups have struggled with a variety of problems associated with overcoming group dynamics to make objective decisions. Most of the problems are associated with the well-documented features of group interaction, where the social and emotional demands on the group often obstruct task-oriented endeavors (Wood & Silver, 1995). The nominal group technique (NGT) is a group decision-making method that has been used to minimize the negative impact of group dynamics and improve the conveyance and convergence of ideas while increasing the likelihood of maximizing participation in, and contribution to the results (Zuech, 1992). The technique, which is particularly useful when groups are required to pool their judgments to decide

among several alternatives (Hornsby, Smith, & Gupta, 1994), has been touted as the "gold standard" for group decision making (Valacich et al. 1994). Researchers have also confirmed that NGT improves creativity and performance in various group decision-making settings (Delbecq et al.. 1975; Van de Ven & Delbecq, 1971).

The process contributes to greater objectivity as it seeks to reduce emotional attachment to ideas by hiding the source of each generated idea (Delbecq & Van de Ven, 1971). It is touted as a useful technique for increasing participation in group discussions (Stephenson et al. 1982). For example, in a study conducted by Gresham (1986) participants were found to react positively to the technique and commented that they were satisfied with its structure in comparison to other previously used group techniques.

The NGT was originated by Delbecq & Van de Ven (1971) and Delbecq et al. (1975) for conducting potentially problematic group meetings. It embodies a set of procedures for conducting group sessions with an independent facilitator. The essence of the "nominal" group is that participants work in close proximity (face-to-face) without interacting initially, and are disassociated from the ideas they contribute in later stages. The process includes a creative thinking and idea generation phase, followed by evaluation and finally decision making. The NGT procedure in its pure form is as follows (Delbecq et al., 1975):

1. Participants independently and silently generate ideas regarding goals and problem solutions in writing.
2. The facilitator records one idea at a time from group members in a round-robin format until all participants have completed their list of ideas.
3. Each idea is discussed for clarification and subsequent evaluation, without either critical evaluation or lobbying.
4. Participants independently rate and rank all the ideas
5. The final decision making on the priority ordering of the alternatives is based on voting and mathematical pooling of the individual rankings.

In many group decision-making sessions, participants are reluctant to suggest ideas because of the fear of looking foolish or being criticized, as well as the reluctance to create conflicts in the group. The division of NGT activities into creative thinking and idea generation, evaluation and then decision making, prevents the domination of the discussion by a single participant and encourages the more passive group members to participate. In essence, it minimizes negative group dynamics that inhibit the free flow of information and opinions. In freely interacting groups, these negative behaviours and dynamics include (1) conforming behavior, (2) anchoring, (3) groupthink, and (4) the Abilene Paradox. *Conforming* behavior refers to a lack of significant deviation from the average, expected behavior or norm, while *anchoring* is a common human tendency to reply too heavily on one piece of information when making decisions. *Groupthink* is a type of thought exhibited by group members who try to minimize conflict and reach consensus without critically testing, analyzing and evaluating ideas. Finally, the *Abilene Paradox* is a situation in which a group of people collectively decide on a course of action that is counter to the preferences of any of the individuals in the group. On the contrary, the NGT promotes a silent and confidential approach to idea generation which enhances creative thinking and by extension can contribute to rich discussion with the likelihood of better solutions (Delbecq et al. 1986; Moore, 1987).

Although the NGT method is easy-to-use and easily adaptable to various problem domains (Chapman, 1998) it has its limitations. It forces the facilitator to establish appropriate timing for each NGT intervention (Duggan & Thachenkary, 2004); it tends not to enhance the building of team rapport (Richmond & McKnelly, 1996); it is not effective in synthesizing ideas (Broome, 1994; MacPhail, 2001) and some facilitators may insist on equal participation by all participants instead of participation based on participant's knowledge and experience (Duggan & Thachenkary, 2004).

Despite these limitations, researchers have confirmed NGT's superiority over freely interacting techniques in cases involving creative problem solving (Delbecq et al., 1986; Van de Ven & Delbecq, 1971, 1974), its effectiveness with heterogeneous groups solving complex problems (Stephenson et al., 1982), and its usefulness in exploring potential obstacles to information systems deployment (Henrich & Greene, 1991). In addition, it has been used in collaboration with other decision-making techniques (Duggan & Thachenkary, 2004; Thomas, et al. 1989; Zuech, 1992).

In a study conducted by Van de Ven & Delbecq (1974) exploring the effectiveness of NGT, Delphi and interacting group decision making techniques, it was found that NGT and Delphi are equally effective and that both techniques are more effective than conventional interacting groups. They discovered that both NGT and Delphi methods can improve group processes and they are highly suited in cases where the pooled judgment of a group of people is required. However, consensus was closer in the NGT than in the Delphi method. In addition, the consensus reached in a Delphi session can be a manipulated consensus, which may not reflect the true feeling

of the group (Mitroff & Turoff, 1975). The procedures in both methods are similar except that the original Delphi uses mailed questionnaires instead of face-to-face meetings.

In summary, research suggests that NGT groups can provide a reasonable balanced participation between group members, more creative ideas, increased perception of accomplishment and greater satisfaction with the quality of ideas generated and a greater level of group efficiency. These benefits support our selection of the NGT as the group decision-making method used in this study, to enable our objective of identifying those SPIs that are most salient to small software development firms in the English-speaking Caribbean, and by extension, our aim to produce a framework that can be feasibly implemented in and used by these firms.

METHODOLOGY

Participants in the NGT sessions were first exposed to the eighteen level 1, 2 and 3 practices identified in the CMMI. Participants were also asked to identify any other SPI practices which they believed were relevant to small firms in the ESC; practices could also be combined, particularly if these were viewed as inseparable. Drawing on the seeding list of 18 practices from the CMMI as well as any new (or combined) practices identified in the session, participants were asked to identify relevant and applicable practices, based on the norms, constraints and culture that prevails in their countries.

In applying the NGT in this study, the creative thinking and idea generation phase of the NGT was modified by seeding the discussion with the list of established software development practices. This decision to modify the NGT process was based on the premise that Software Engineering Institute (SEI) has established SPI practices that have been tried, tested and proven in the IS domain for many years and have been found to be useful. It was therefore believed that exposing participants to these well established practices at the beginning of the process would help enhance and stimulate creative thinking and the idea generation phase of the NGT process. Although it is acknowledged that seeding the discussion could also result in some bias towards these 18 Levels 1-3 practices, participants were encouraged to add 'new' practices or combine existing practices if they believed these representations were the most relevant. Hence, the modified NGT enabled two important aspects of the study: generation of new ideas and a possible set of outcomes that could be clearly linked with existing practices. This in turn made possible a key goal of this study: to produce a framework that can be related to established process improvement frameworks – in this case, the popular and leading CMMI.

The modified NGT approach taken was as follows:

1. Idea Selection (Formerly called ‘Brainstorming’ in the pure NGT process)
    (a) Participants were given a list of all established 18 CMMI Process Areas (practices)
    (b) On 3” x 5” cards participants were asked to silently (i) select or merge any of the 18 CMMI practices or (ii) add in 'new' practices, deemed necessary and relevant within the ESC context
2. Round-robin of ideas
    (a) Cards were collected by the facilitator, shuffled and randomly returned to participants
    (b) Participants were asked to verbally state one idea per card in a round-robin manner (The purpose of shuffling the cards was to hide the identity of the generator of each idea)
3. Clarification of ideas
    (a) Participants were asked to explain the thought and meaning of new or merged practices; no defense was necessary
    (b) Duplicates (practices) were eliminated
4. Select the top ten Practices
    (a) Participants were asked to select the top ten practices
    (b) These were recorded and tallied
    (c) The ten most frequent practices were selected
5. Ranking
    (a) Participants were asked to silently rank these ten practices, with the most important practices being given a score of 10 and the least a score of 1
    (b) The scores were recorded and tallied
6. Decision making
    (a) The practice with the highest score was ranked as number 1, the next highest score as number 2, etc.
    (b) The final list was presented to the group for their comment and approval

Three pilot sessions were held in Jamaica using the modified NGT approach to identify the key SPI practices in the ESC. Participants in these three pilot sessions were mature PhD students and IT staff at the University of the West Indies who have been involved in software development projects. Adjustments were made to the NGT process as well as adjustments to the design of the post evaluation questionnaires that both the facilitator and

participants were asked to complete. Because all three pilot sessions were held in Jamaica we considered this the 'Jamaican Experience'

Four main data collection sessions were conducted in Guyana, Trinidad, Barbados and Jamaica in April 2008 with IS experts from each country. Our justification for selecting these four countries is based on the fact that they have established software development firms and account for eighty-three percent (83%) of the ESC population. There were a total of thirty participants from the four countries who worked in software development firms in IS capacities or team members on IS projects. On average the NGT sessions ran for approximately 2 hours. The composition of the participants is shown in Table 1.

Table 1. Composition of Participants

Job Title	**Number of Participants**
Analyst/developer	16
Senior systems analyst/Chief engineer	7
Senior manager/Managing consultant	7
Total	**30**

Participants were asked to identify relevant and applicable software development practices which they considered important, based on the norms and constraints in their respective countries. They were also given instructions to merge any process areas that they felt were necessary based on their knowledge and experience.

FINDINGS

At the end of the four NGT sessions, twenty-four practices were identified and ranked. This total of twenty-four practices was higher than the original established eighteen (18) because a few practices were combined and new ideas introduced. This supports the notion that the NGT process (in its modified form) can facilitate the generation of more creative ideas. Following the four sessions in which individuals ranked the various practices, we aggregated the scores to tally the top ranked practices in each of the four countries - Guyana, Trinidad, Barbados and Jamaica. In this paper, the twelve top ranked practices in each country are reported (Table 2). A score of 12 was given to the top ranked practices, and then a score of eleven for the next highly ranked, then ten and so forth as shown in Table 2.

For example, Risk Management (RSKM) was ranked as number one in Guyana, then Project Planning (PP), then Requirement Development (RD), and so these were given scores of twelve, eleven and ten respectively. Likewise, in Trinidad and Jamaica the top ranked practice was Requirements Development merged with Requirements Management (RD + RM). Based on the approach taken, this practice was given a score of twelve. The same approach was taken at Barbados where the top ranked practice was Requirement Management merged with Organization Process Definition (RM + OPD) which was given a score of twelve.

Table 2. Result in each country

Guyana		**Trinidad**		**Barbados**		**Jamaica**	
RSKM	12	RD+RM	12	RM+OPD	12	RD+RM	12
PP	11	TS	11	RSKM	11	IPM+PP+PMC	11
RD	10	IPM	10	OT	10	OT	10
TS	9	RSKM	9	PP+PMC	9	RSKM	9
IPM	8	VER+VAL	8	TS	8	TS	8
PMC	7	OT	7	OPF	7	OPD	7
OT	6	PMC	6	RD	6	OPF	6
VER+VAL	5	PP	5	OPD	5	PI+VER+VAL	5
OPD	4	OPF	4	IPM	4	PPQA	4
VER	3	OPD	3	PP	3	VER	3
OPF	2	CM	2	RM	2	M&A	2
VAL	1	PI	1	DAR	1	PI+VAL	1

Upon completion of this exercise, tallying was done to ascertain the highly ranked practices in the four countries. For example, Risk Management (RSKM) received scores of 12, 9, 11 and 9 in Guyana, Trinidad, Barbados and

Jamaica respectively, giving a grand total of 41. The next practice with the highest score was Technical Solution (TS), receiving a grand total of 36.

The partial list of the tallying result is shown in Table 3, outlining the top fourteen practices with their associated scores and their established CMMI maturity levels.

Table 3: The Top Ranked Practices in Descending Order

No.	Practices	Score	CMMI Maturity Level
1	RSKM	41	3
2	TS	36	3
3	OT	33	3
4	RD+RM	24	2 & 3
5	IPM	22	3
6	PP	19	2
7	OPD	19	3
8	OPF	19	3
9	RD	16	3
10	PMC	13	2
11	VER+VAL	13	3
12	RM+OPD	12	2 & 3
13	IPM+PP+PMC	11	2 & 3
14	PP+PMC	9	2

In computing the final set of top ranked key software development practices, practice #9 – Requirements Development (RD) was deleted from the initial top ranked list because it was already included in Practice #4 – Requirements Development + Requirements Management which are concerned with meeting the user's expectation. This action elevated Practice #11, VER + VAL (verification and validation) a set of quality assurance practices, into Position #10; neither practice was represented in the higher ranked practices, hence this was retained in the set of top practices. A review of the remaining practices (i.e. #12 to #14) showed that all other merged practices were already incorporated in the first ten highly ranked practices. For example, for RM + OPD, RM is addressed in Practice #4 and OPD is addressed in Practice #7. Practices 12-14 were therefore dropped from the final list of 'most relevant' SPI practices. Although the final number of practices was not predetermined in this study, the refinement process yielded a list of ten highly ranked practices (the "Top Ten") for small software development firms.

Table 4 presents the resulting framework of the Top Ten ranked practices with their score, designation and assigned maturity levels. The original CMMI framework has eighteen practices in levels 2 and 3. However, the resultant framework was more simplified converging on only ten practices – three assigned to level 2 and seven assigned to levels 3. Assigned levels were guided by the CMMI approach in which level 2 practices deal with project management and level 3 practices being concern about organizational wide process management.

Table 4: The Resulting Simplified Framework

Rank	Practices	Score	Designation	Assigned Maturity Level
1	RSKM	41	Risk Management	3
2	TS	36	Technical Solution	3
3	OT	33	Organization Training	3
4	RD+RM	24	Requirements Development & Requirements Management	2
5	IPM	22	Integrated Project Management	3
6	PP	19	Project Planning	2
7	OPD	19	Organization Process Definition	3
8	OPF	19	Organization Process Focus	3
9	PMC	13	Project Monitoring & Control	2
10	VER+VAL	13	Verification & Validation	3

On the subject of the NGT process, in general good outcomes were received in all four sessions. Based on the facilitator's feedback, only 15% of the participants did not contribute to the group decision (4 out of 30). The inverse gives an 87% participation rate. The participants failed to agree on only one of the suggestions mooted

regarding merging multiple practices. In general, the facilitator was also very satisfied that participants treated the exercise seriously.

DISCUSSION

The results of this study yielded a more simplified process improvement model for levels 1 – 3, with a 45% reduction (from 18 to 10) in the number of practices in comparison to the original CMMI framework. This simplified framework has no practices at level 1 which is consistent with the CMMI framework, in which there are no structured practices at this level. Because we are taking a phased approach in developing this model, there are currently no practices at levels 4 and 5. This approach is consistent with our goal to propose a model that is more relevant and indeed easier to implement in small firms. In time, we expect that firms will gradually incorporate some advanced practices at levels 4 and 5 when learning and institutionalization has taken place.

In examining the top three highly ranked practices (which are all level 3 practices) – risk management, technical solution and organization training, it is evident that software developers in the ESC are generally more concerned with wider organizational process management issues. These include action planning to overcome project exposures, finding a solution that meets users requirements and equipping the project team members with the necessary knowledge so that they can execute their tasks in the most efficient manner. This claim is made based on the aims of these practices as distilled by SEI in their CMMI framework. The aims are:

1. Risk management - To identify potential problems before they occur so that risk-handling activities can be planned and invoked as needed across the life of the product or project to mitigate adverse impacts on achieving objectives
2. Technical solution – To design, develop, and implement solutions to requirements
3. Organization training – To develop the skills and knowledge of people so that they can perform their roles effectively and efficiently

The higher ranking of practices related to risk management, technical solution and organization training may also reflect the firms' focus in light of limited resources in finance, labor, equipment and material that are available in the ESC.

In reviewing the limitations of our study, the use of convenience sampling to select participants for the sessions is a limiting factor. However, this approach is a result of the limited pool of available IS experts in the ESC to assist with such an exercise. The International Monetary Fund reports that Jamaica loses 70% of its tertiary graduates annually through migration (Gleaner, 2006). Another limitation stems from the fact that only four of the seventeen ESC countries were surveyed. Although only four countries were surveyed, these represented 83% of the software development firms in the ESC. Nonetheless, other software development firms in other ESC countries could have been included to better support the claim that the outcomes reflect the 'English-Speaking Caribbean Experience'.

CONCLUSION

The application of the NGT process allowed us to gain rapid convergence on a set of key software practices that can be used as a basis for proposing a modified SPI framework, for use in ESC countries and other developing countries with similar profiles.

Qualitatively, we experienced in the sessions, the much touted claim of the effectiveness of NGT in the stimulating idea conveyance and for enabling rapid convergence towards a decision. This study confirms the notion that NGT groups make possible a reasonable balanced participation among members, the generation of more creative ideas, and an increased perception of accomplishment and a high level of satisfaction among participants.

With the use of the NGT process, we were able to identify the applicable SPI practices in the ESC. The outcome yielded a simplified process improvement framework incorporating ten SPI practices that were considered most salient to small software development firms. We hope that software development firms in the ESC will perceive the simplified framework as less cumbersome, disruptive and costly to implement. It is further hoped that the usage of our simplified framework will increase the likelihood of producing higher quality software products in the ESC. It is equally hoped that other researchers will validate the framework and possibly modify the framework to support higher levels of quality impact.

REFERENCE

Batista, J., & de Figueiredo, A. D. (2000). SPI in a Very Small Team: a Case with CMM. *Software Process Improvement and Practice, 5*, 243-250.

Beecham, S., Hall, T., & Rainer, A. (2005). Defining a Requirements Process improvement Model. *Software Quality Journal, 13*, 247-279.

Bicego, A., & Kuvaja, P. (1996). Software Process Maturity and Certification. *Journal of Systems Architecture, 42*(9), 611-620.

Broome, B. (1994). *The role of facilitated group process in community-based planning and design*. New Jersey: Hampton Press.

Buchman, C. D., & Bramble, L. K. (1995). Three-tiered Software Process Assessment Hierarchy. *Software Process Improvement and Practice, 1*(2), 99-106.

Chapman, R. J. (1998). The effectiveness of working group risk identification and assessment techniques. *International Journal of Project Management, 16*(6), 333-343.

Chevers, D. A., & Duggan, E. W. (2007). A Modified Capability Framework for Improving Software Production Processes in Jamaican Organizations. *The Electronic Journal on Information Systems in Developing Countries, 30*(4), 1-18.

Delbecq, A. L., & Van de Ven, A. H. (1971). A group process model for problem identification and program planning. *Applied Behavioural Science, 7*, 466-491.

Delbecq, A. L., Van de Ven, A. H., & Gustafson, D. H. (1975). *Group techniques for program planning: A guide to nominal group and Delphi processes*. Glenview, Illinois: Scott, Foresman & Company.

Delbecq, A. L., Van de Ven, A. H., & Gustafson, D. H. (1986). *Group Techniques for Program Planning: A Guide to Nominal Group and Delphi Processes*. Middleton, WI: Greenbriar.

Deming, W. E. (1986). *Out of the Crisis*. Cambridge, MA.: MIT Center for Advanced Engineering.

Dooley, K., Subra, A., & Anderson, J. (2001). Maturity and its impact on new product development project performance. *Research in Engineering Design, 13*(1), 23-29.

Duggan. (2003). Silver Pellets for Improving Software Quality. *Information Resources Management Journal, 17*(2), 1-21.

Duggan, & Thachenkary, C. S. (2004). Integrating Nominal Group Technique and Joint Application Development for Improved Systems Requirements Determination. *Information and Management, 41*(4), 399-411.

Duggan, & Virtue, G. (2004). Bridging the Digital Divide in Caribbean Group Decision-Making. *The Electronic Journal of Information Systems in Developing Countries, 17*(5), 1-23.

El Emam, K., & Madhavji, N. H. (1995). *The Reliability of Measuring Organizational Maturity*: John Wiley & Sons.

Gleaner. (2006, February 20, 2006). Major Brain Drain. *Jamaica Daily Gleaner (The)*.

Gresham, J. (1986, December 5, 1966). *Expressed satisfaction with the nominal group technique among change agents*. Paper presented at the Proceedings of the 13th Annual National Agricultural Education Research Meeting, Dallas, Texas.

Harter, Slaugher, S. A., & Krishnan, M. S. (2000). Effects of process maturity on quality, cycle time, and effort in software product development. *Management, 46*(4), 451-466.

Hasse, V. H. (1996). Software Process Assessment Concepts. *Journal of Systems Architecture: the EUROMICRO Journal, 42*(9), 621-631.

Henrich, T. R., & Greene, T. J. (1991). Using the nominal group technique to elicit roadblocks to MRP ll implementation. *Computers & Industrial Engineering, 21*, 335-338.

Herbsleb, J., Zubrow, D., Goldenson, D., Hayes, W., & Paulk, M. (1997). Software Quality and the Capability Maturity Model. *Communications of ACM, 40*(6), 30-40.

Herrera, E. M., & Ramirez, R. A. J. (2003). A Methodology for Self-Diagnosis for Software Quality Asssurance in Small and Medium-sized Industries in Latin America. *The Electronic Journal of Information Systems in Developing Countries, 15*(4), 1-13.

Hornsby, J. S., Smith, B. N., & Gupta, J. N. (1994). The impact of decision-making methodology on job evaluation outcomes. *Group & Organization Management, 19*, 112-128.

Humphrey. (1989). *Managing the Software Process*. Reading, MA.: Addison-Wesley.

Johnson, D. L., & Brodman, J. G. (1996). Realities and Rewards of Software Process Improvement. *IEEE Software, 13*(6), 1-5.

Krishnan, M. S., & Keller, M. I. (1999). Measuring Process Consistency: Implications for Reducing Software Defects. *IEEE Transactions On Software Engineering, 25*(6), 769-781.

MacPhail, A. (2001). Nominal group technique: a useful method for working with young people. *Britisj Educational Research Journal, 27*(2), 161-170.

Mitroff, I. I., & Turoff, M. (1975). *The Delphi method: Techniques and applications*. Reading MA: Addison-Wesley.

Moore, C. M. (1987). *Group Techniques for Idea Building*. Newbury Park, CA: Sage.

Ngwenyama, O., & Nielsen, P. A. (2003). Competing Values in Software Process Improvement: An Assumption Analysis of CMM from an Organizational Culture Perspective. *IEEE Transactions on Engineering Management, 50*(1), 100.

Paulk, Curtis, B., Chrissis, M. B., & Weber, C. V. (1993). Capability Maturity Model for Software Version 1.1. *(Tech. Rep. No. CMU-SEI-93-TR-24). Software Engineering Institute*.

Paulk, Weber. C.V., Curtis, B., & Chrissis, M. B. (1995). *The Capability Maturity Model: Guidelines for Improving the Software Process*. Reading, Massachusetts: Addison Wesley Longman, Inc.

Persse, J. R. & Wiley, J. (2001). Implementing the Maturity Model. *The TQM Magazine, 14*(2), 79-91.

Ravichandran, T., & Rai, A. (2000). Quality Management in Systems Development: An Organizational System Perspective. *MIS Quarterly, 24*(3), 381-415.

Reifer, D. J. (2000). The CMM: It's formidable. *Journal of Systems and Software, 50*, 97-98.

Richmond, E. B., & McKnelly, M. T. (1996). Alternative user survey and group methods - nominal group technique applied to US depository libraries. *Journal of Government Information, 23*(2), 137-149.

Rogoway, P. (1998). How to Reap the Business Benefit from SPI: Adding SPICE while preserving the CMM (Motorola). *SPI Newspaper*.

SEI. (2004). Upgrading from SW-CMM to CMMI. *http://www.sei.cmu.edu/cmmi/adoption/pdf/upgrading.pdf*.

SEI. (2006). Capability Maturity Model Integration (CMMI) Version 1.2 Overview. *http://www.sei.cmu.edu/cmmi/adoption/pdf/cmmi-overview07.pdf*, 7.

Stephenson, B. Y., Michaelsen, L. K., & Franklin, S. G. (1982). An empirical test of the nominal group technique in state solar energy planning. *Group & Organization Studies, 7*, 320-334.

Thomas, J. B., McDaniel, R. R., & Dooris, M. J. (1989). Strategic issue analysis: NGT + decision analysis for resolving strategic issues. *Journal of Applied Behavioural Science, 25*, 189-200.

Thong, J. Y. L., Yap, C., & Raman, K. S. (1996). Top Management Support, External Expertise and Information Systems Implementation in Small Businesses. *Information Systems Research, 7*(2), 248-267.

Turner. (2007). CMMI Mentoring Service. *http://www.tcg.com/services-CMMI-mentoring-overture.html?OVRAW=elements%20*.

Valacich, J., Dennis, A. R., & Connolly, T. (1994). Idea generation in computer-based groups: A new ending to an old story. *Organizational Behaviour and Human Decision Processes, 57*(3), 448-468.

Van de Ven, A. H., & Delbecq, A. L. (1971). Nominal versus interacting group processes for committee decision-making effectiveness. *Academy of Management Journal, 14*(2), 203-212.

Van de Ven, A. H., & Delbecq, A. L. (1974). The effectiveness of nominal, Delphi, and interacting group decision making processes. *Academy of Management Journal, 17*, 605-621.

Weber, C., & Layman, B. (2002). Measurement Maturity and the CMM: How Measurement Practices Evolve as Process Mature. *Software Quality Professional, 4*(3), 6-20.

Wood, J., & Silver, D. (1995). *Joint Application Development*. New York: John Wiley & Sons.

Zuech, N. (1992). Identifying and ranking opportunities for machine vision in a facility. *Industrial Engineering, 24*, 42-44.

COPYRIGHT